AD-A049 287 CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER --ETC F/G 17/2
WORD HYPOTHESIZATION FOR LARGE-VOCABULARY SPEECH UNDERSTANDING --ETC(U)
OCT 77 A R SMITH F44620-73-C-0074

UNCLASSIFIED AFOSR-TR-78-0005 NL

1 OF 2
AD
A049287

Word Hypothesization for Large-Vocabulary Speech Understanding Systems

A. Richard Smith

October 20, 1977

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

Word Hypothesization for Large-Vocabulary Speech Understanding Systems

A. Richard Smith

October 20, 1977

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA, 15213

Submitted to Carnegie-Mellon University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science.

This work is supported in part by the Defense Advanced Research Projects Agency under contract number F44620-73-C-0074 and is monitored by the Air Force Office of Scientific Research.

Abstract

This thesis describes research directed toward the development of general English speech understanding systems. The relatively unconstrained grammars and large vocabularies characterizing such systems require them to eliminate most of the words found in their vocabularies by using only acoustic information. In particular, we present the design and performance of a bottom-up word hypothesizer capable of handling large vocabularies (>10,000 words) which takes segmented and labeled speech as input and produces word hypotheses. The primary concerns of the thesis are the problems involved with large vocabularies and the effect of large vocabularies on word hypothesization.

The thesis deals with the following problems: 1) Knowledge Representation: storing the acoustic knowledge of words efficiently for fast retrieval; 2) Knowledge Acquisition: obtaining the acoustic knowledge for a large number of words easily; 3) Flexibility: permitting improvements to be made to the acoustic processers of the speech system (e.g., segmenter-labeler) without requiring an expensive reacquisition of knowledge; and 4) Performance: hypothesizing many of the correct words of an utterance and few incorrect ones within "reasonable" computation constraints.

The solutions to these problems center around the knowledge representation used by the word hypothesizer. Speech is represented in a hierarchy of levels containing (from bottom to top): segment labels, sylparts, syllables, and words. (Sylparts include onsets (the initial non-nucleus part of a syllable), vowels and codas (the final non-nucleus part of a syllable).) Knowledge is stored in a hierarchy-tree representation. That is, between each pair of adjacent levels (segment-sylpart, sylpart-syllable, and syllable-word level pairs) is a tree structure storing a sequence of lower level units to define a higher level unit. The tree between each pair of levels permits merging common initial parts of sequences to reduce storage costs and recognition time.

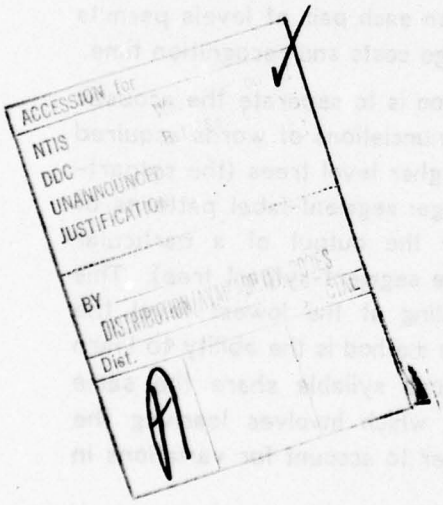
The solution to the problem of knowledge acquisition is to separate the acoustic description of words into a) a priori knowledge: base pronunciations of words acquired from a word-phoneme dictionary and stored in the two higher level trees (the sylpart-syllable and syllable-word trees) and b) learned knowledge: segment-label patterns of the sylparts acquired by training the hypothesizer on the output of a particular segmenter-labeler and stored in the lowest level tree (the segment-sylpart tree). This solution is made possible by several methods of handling at the lowest level the coarticulation problems common in continuous speech. One method is the ability to learn a vowel-sequence, which may occur when more than one syllable share the same syllable nucleus. A second method is context-learning, which involves learning the surrounding segment-context of a segment-pattern in order to account for variations in the segment-patterns learned for a sylpart.

We present several measures in order to evaluate the storage and recognition cost efficiency of the representation, analyze the recognition algorithm, and evaluate the performance of the word hypothesizer over different vocabulary sizes.

The word hypothesizer is tested on 105 utterances (705 words) for 7 different vocabulary sizes ranging from 500 words to 19,000 words. The performance for these vocabularies ranges from a word accuracy of 73% at an average rank of 2.6 for the correct hypotheses using a 500-word vocabulary to a word accuracy of 58% at an average rank of 5.8 using a 19,000-word vocabulary. According to the average efficiency measure (developed here), this performance degrades at approximately a logarithmic rate over the range of vocabulary sizes tested. The computation costs begin at 2.4 MIPSS (million of instructions per second of speech) for the 500-word vocabulary and increases at a logarithmic rate to 6.6 MIPSS for the 19,000-word vocabulary.

We conclude that bottom-up word hypothesization is not greatly effected by the size of the vocabulary and that with improvements in the word hypothesizer and the segmenter-labeler, speech understanding systems for general English can obtain a great amount of constraint from the acoustics alone.

The major contributions are: 1) A better understanding of the effect of large vocabularies for speech understanding systems, 2) A solution to the problem of knowledge acquisition for an AI knowledge-based system, 3) Several methods for handling at low representation levels of speech some of the coarticulation problems of continuous speech, and 4) The design of a bottom-up word hypothesizer that performs better than earlier word hypothesizers.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AFOSR-TR-78-0005	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): WORD HYPOTHESIZATION FOR LARGE-VOCABULARY SPEECH UNDERSTANDING SYSTEMS.	5. TYPE OF REPORT & PERIOD COVERED: Interim rept.
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): A. Richard Smith	8. CONTRACT OR GRANT NUMBER(s): F44620-73-C-0074
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Carnegie-Mellon University, Computer Science Dept., Pittsburgh, PA 15213	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F, 2304/A2, A2
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Advanced Research Projects Agency, 1400 Wilson Blvd, Arlington, VA 22209	12. REPORT DATE: October 20, 1977
	13. NUMBER OF PAGES: 124
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Air Force Office of Scientific Research/NM, Bolling AFB, DC 20332	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis describes research directed toward the development of general English speech understanding systems. The relatively unconstrained grammars and large vocabularies characterizing such systems require them to eliminate most of the words found in their vocabularies by using only acoustic information. In particular, we present the design and performance of a bottom-up word hypothesizer capable of handling large vocabularies (>10,000 words) which takes segmented and labeled speech

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403 081

20. as input and produces word hypotheses. The primary concerns of the thesis are the problems involved with large vocabularies and the effect of large vocabularies on word hypothesization.

The thesis deals with the following problems: 1) Knowledge Representation: storing the acoustic knowledge of words efficiently for fast retrieval; 2) Knowledge Acquisition: obtaining the acoustic knowledge for a large number of words easily; 3) Flexibility: permitting improvements to be made to the acoustic processers of the speech system (e.g., segmenter-labeler) without requiring an expensive reacquisition of knowledge; and 4) Performance: hypothesizing many of the correct words of an utterance and few incorrect ones within "reasonable" computation constraints.

The solutions to these problems center around the knowledge representation used by the word hypothesizer. Speech is represented in a hierarchy of levels containing (from bottom to top): segment labels, sylparts, syllables, and words. (Sylparts include onsets (the initial non-nucleus part of a syllable), vowels and codas (the final non-nucleus part of a syllable).) Knowledge is stored in a hierarchy-tree representation. That is, between each pair of adjacent levels (segment-sylpart, sylpart-syllable, and syllable-word level pairs) is a tree structure storing a sequence of lower level units to define a higher level unit. The tree between each pair of levels permits merging common initial parts of sequences to reduce storage costs and recognition time.

The solution to the problem of knowledge acquisition is to separate the acoustic description of words into a) a priori knowledge: base pronunciations of words acquired from a word-phoneme dictionary and stored in the two higher level trees (the sylpart-syllable and syllable-word trees) and b) learned knowledge: segment-label patterns of the sylparts acquired by training the hypothesizer on the output of a particular segmenter-labeler and stored in the lowest level tree (the segment-sylpart tree). This solution is made possible by several methods of handling at the lowest level the coarticulation problems common in continuous speech. One method is the ability to learn a vowel-sequence, which may occur when more than one syllable share the same syllable nucleus. A second method is context-learning, which involves learning the surrounding segment-context of a segment-pattern in order to account for variations in the segment-patterns learned for a sylpart.

We present several measures in order to evaluate the storage and recognition cost efficiency of the representation, analyze the recognition algorithm, and evaluate the performance of the word hypothesizer over different vocabulary sizes.

The word hypothesizer is tested on 105 utterances (705 words) for 7 different vocabulary sizes ranging from 500 words to 19,000 words. The performance for these vocabularies ranges from a word accuracy of 73% at an average rank of 2.6 for the correct hypotheses using a 500-word vocabulary to a word accuracy of 58% at an average rank of 5.8 using a 19,000-word vocabulary. According to the average efficiency measure (developed here), this performance degrades at approximately a logarithmic rate over the range of vocabulary sizes tested. The computation costs begin at 2.4 MIPSS (million of instructions per second of speech) for the 500-word vocabulary and increases at a logarithmic rate to 6.6 MIPSS for the 19,000-word vocabulary.

20.

We conclude that bottom-up word hypothesization is not greatly effected by the size of the vocabulary and that with improvements in the word hypothesizer and the segmenter-labeler, speech understanding systems for general English can obtain a great amount of constraint from the acoustics alone.

The major contributions are: 1) A better understanding of the effect of large vocabularies for speech understanding systems, 2) A solution to the problem of knowledge acquisition for an AI knowledge-based system, 3) Several methods for handling at low representation levels of speech some of the coarticulation problems of continuous speech, and 4) The design of a bottom-up word hypothesizer that performs better than earlier word hypothesizers.

Acknowledgments

I would like to acknowledge the aid and support of several people who in many ways made the completion of these graduate years possible. First, I would thank my advisor, colleague, and friend, Lee Erman. Combined in one person are all the ideal qualities of an advisor: excellent advice, interest in the student and project, availability, and editing skills. I want to thank Raj Reddy for his guidance in my earlier graduate years and for attracting me to speech understanding research. My thanks also go to the other members of my committee, Jon Bentley and Ron Cole, for their help in improving the quality and readability of the thesis.

At various stages of the work, I received the competent assistance of Jim Berenbaum, Rajiv Bhalla, and Eric Grant. They professionally did what was often tedious work. The environment of the department contributed in the form of excellent facilities and a pool of willing experts.

I received much needed moral support from several people. I want to acknowledge a fellow graduate, Dale Partin, whose "every Monday lunch meetings" helped me to keep a proper perspective on work and life. I would thank my parents for their years of encouragement and prayers. The general interest and expressed concern of many others: friends, relatives, and office-mates, was appreciated. But above all, I publicly acknowledge the encouragement, sympathy, exhortation, and sustaining confidence of my wife Penny. She has successfully brought a husband through a thesis and a baby through the first eight months of life.

Finally, I give ultimate credit to "the One perfect in knowledge" [Job 36:4], *soli Deo gloria.*

Contents

Chapter 1: Introduction

1.1 Motivation

Speech Understanding Systems to date have depended on a very constrained syntax and semantics model of language and a small (≤1000) word vocabulary in order to handle the problem of understanding continuous speech. Though we cannot claim that such systems have solved the continuous speech problem even for such constrained tasks, we should begin to look ahead towards speech systems capable of handling general English with its weakly constrained syntax and semantics and its relatively unlimited vocabulary. One of the characteristics of these more ambitious systems will be their ability to eliminate most of the words found in their vocabularies as possible candidates for what was spoken by using only the acoustic information. It is the purpose of this thesis to investigate to what degree a word hypothesizer can use the acoustic information of the utterance to reduce the search space of possible word sequences for such speech systems. The primary concern will be the effect that large vocabularies have on the performance of a word hypothesizer. The investigation is carried out by designing and implementing a word hypothesizer capable of handling large vocabularies in order to study the effect of vocabulary size.

The work for this thesis actually includes the design and implementation of two word hypothesizers: 1) the POMOW word hypothesizer [Smith - 1976] used in the Hearsay-II speech understanding system, which was demonstrated on September 8, 1976, at Carnegie-Mellon University [CMU Computer Science Speech Group - 1977] and 2) the Noah[1] word hypothesizer that developed out of POMOW and which was designed specifically with large vocabularies in mind. POMOW will be considered here as previous research and will be looked at only to understand some of the problems which had to be handled by Noah. This chapter discusses the word hypothesization problem, looks at previous research, and gives an overview of Noah and the remaining chapters.

---

1 For those interested in name derivations, "POMOW" was derived from "Phones hypOthesize Morphemes (syllables) hypOthesize Words". "Noah" was named for Noah Webster, an early American lexicographer.

1.2 The Problem

The nature of speech is such that there is no direct mapping from acoustic information to a unique spoken word. The acoustic pattern of a word is embedded within the total pattern of the utterance and modified by it. This is called the coarticulation problem. A listener interprets an acoustic event not only by what actually occurs in the utterance but also by the surrounding context and even by what he expects to hear. Environmental noise, differences between speakers, differences for the same speaker at different times, and variations in pronunciations also add to the difficulty of finding what words were spoken in an utterance. Another problem is carelessness by the speaker; it seems that a person often speaks just well enough to be understood (most of the time) by another human [Newell - 1975].

1.2.1 Background of Speech Recognition and Understanding Research

The history of research toward a solution of the above problems is a history of a step by step relaxing of constraints on the problem as progress was made. The work of Reddy and Reddy & Vicens at Stanford University ([Reddy - 1966]; [Reddy and Vicens - 1968]; [Vicens - 1969]) resulted in extending the state-of-the-art of isolated word recognition systems, (e.g., 91% accuracy on a 561-word vocabulary in ten times real-time on a PDP10 and with live input). Although the Vicens-Reddy system increased the vocabulary size by an order of magnitude over that permitted by former word recognition systems, coarticulation problems were avoided by requiring a user to speak the words of a sentence separated by short silences. Important differences between that system and earlier ones were that it contained a substantial amount of speech knowledge and it used extensive heuristics in applying the knowledge to prune the search space. Early word recognition systems were essentially pattern classifiers.

The Hearsay-I model of speech understanding, developed at CMU during 1970-1971 ([Reddy, Erman, and Neely - 1972]), faced the problems of connected speech. The implementation of this model as the Hearsay-I system [Reddy, Erman, and Neely - 1973], resulted in the first demonstrable (June, 1972) live system to handle non-trivial connected speech. In order to handle connected speech and obtain a sentence and word accuracy of 79% and 93%, respectively, the system depended on a very constrained syntax and semantic model of speech (e.g., the chess task) and a very small vocabulary (<40 words). The system served to clarify the nature and necessary interaction of several sources of knowledge by using three independent cooperating sources of knowledge (acoustic-phonetic, syntactic, and semantic).

Concurrent with the development of the Hearsay-I model, a group was formed by the Advanced Research Projects Agency (ARPA) to study the feasibility of developing speech understanding systems. The resulting report [Newell, et al. - 1971]

gives a comprehensive and detailed analysis of the problems involved and specifies reasonable constraints for a five year research effort on the problem. Notable among these constraints are: [The system should] "accept connected speech,..., permitting a slightly selected vocabulary of 1000 words, with a highly artificial syntax, and a task with a constrained and fairly simple semantics, ..., tolerating less than 10% semantic error, in a few times real-time, ...".

On recommendation of the study group a five year ARPA Speech Understanding Research effort began in October, 1971. Of the speech systems demonstrated in 1976, the Harpy system [Lowerre - 1976] succeeded in meeting the specifications of the project; the Hearsay-II system came close.

Common to all of the speech systems resulting from the five-year effort (several of which are mentioned below) is the use of the hypothesize-and-test paradigm. That is, these systems attempt to solve the speech understanding problem by an iterative process of a) creating hypotheses, "educated guesses" about some aspect of the problem, and b) testing the plausibility of the hypotheses. An important part of these systems is word "guessing", i.e., word hypothesizing.

1.2.2 Top-Down versus Bottom-Up Word Hypothesization

Faced with the problem of finding the acoustic pattern of a word embedded within the total pattern of the utterance, researchers have often taken the approach of avoiding any word hypothesization from the lower acoustic levels[2]. Rather, all hypothesization is based on syntactic constraints from higher levels. Word <u>verification</u> (the <u>test</u> part of the hypothesize-and-test paradigm) is then performed by generating a low-level representation (such as phones[3], acoustic segments, or even a spectrogram) for each hypothesized word and then matching this representation against a part of the actual input to derive a score. The word with the best score is taken to be the spoken word for that part of the utterance.

One method of doing top-down word-hypothesization / word-verification is found in Hearsay-I. The system works left-to-right (or right-to-left) through the complete utterance. All words that can legally begin (end) a sentence are hypothesized at the beginning (ending). Those words rated well by the word verification step give

---

2 Speech is often viewed by systems as having different representations in a hierarchy of levels beginning at the bottom with the speech waveform, continuing up through levels such as parametric, phonetic, syllabic, lexical (or word), and ending in a phrase or semantic level at the top.

3 We will use "phone" to refer to a sound detected and classified by a program and "phoneme" for the expectation of that sound as entered in a word pronunciation dictionary.

syntactic and semantic constraint for hypothesizing the following (preceding) adjacent word. This process is continued across the utterance until the end (beginning) is reached.

Such top-down methods of word hypothesization (also called analysis-by-synthesis) are potentially more accurate than the alternative bottom-up hypothesization from the acoustics. This is because the transformation and match takes place for each word with knowledge about the possible context around the word and with knowledge about how various acoustic events within the word might interact to produce the speech. Linguists have developed a fairly detailed generative model of speech describing these transformations from words to the speech signal. The problem with top-down methods is that they are slow if many words must be matched in each place in the utterance; systems waste time matching words which may be syntactically correct but have very little acoustic support. As the semantic and syntactic speech model becomes more general, top-down systems are swamped by the number of words hypothesized. Hearsay-I [Reddy, Erman, and Neely - 1973], the Lincoln Lab Speech System [Forgie - 1974], the Harpy Speech System [Lowerre - 1976] and the IBM Speech Recognition System [Bahl, et al. - 1976] are examples of systems using top-down methods.

The method of bottom-up word hypothesization attempts to infer from the acoustic information what subset of words from the vocabulary may have been spoken in each part of the utterance. These hypotheses may then be verified as in the top-down methods.

The amount of acoustic information used for word hypothesization[4] varies. One rarely-used version of Hearsay-I tried to use gross acoustic events, such as the "SH" in "bishop", to suggest possible words in a region. We know of only two examples of word hypothesizers using extensive acoustic information. They are Klovstad's "Probabilistic Lexical Retrieval Component" for the BBN (Bolt, Beranek and Newman Inc.) HWIM system [Klovstad - 1976] and POMOW [Smith - 1976], the word hypothesizer knowledge source in Hearsay-II.

Although the HWIM system [Woods, et al. - 1976] is oriented towards a top-down approach of speech recognition, in one mode of operation of the system the lexical retrieval component does an initial scan of the utterance to find the best N words (N=15 currently) which fit in the utterance. Assuming that the highest rated words are correct, the system tries to match words which are syntactically consistent to the left and/or to the right of each hypothesized word. Continuing in this way, the grammar

---

4 In the remainder of the thesis, the term "word hypothesization" by itself will be used to mean bottom-up word hypothesization

controls what words are matched adjacent to each highly rated word until the utterance is covered. The lexical retrieval component is driven by the possible phone sequences hypothesized for the utterance. For each partial sequence of phones, the probability for each of the best words matching the sequence is computed. The best performance of the HWIM system, however, is not obtained using that mode, but rather with a left-to-right, top-down control (i.e., similar to the approach used by Hearsay-I).

The POMOW word hypothesizer uses segment-label sequences to hypothesize many words throughout the utterance for the Hearsay-II speech system. The segment level description of speech is obtained by segmenting the acoustic description into similar parts (e.g., silence-like, nasal-like, noise-like, etc.). Each part is characterized by assigning it a label (hence, segment-labels). The knowledge source that does this work is called the segmenter-labeler. In general, the segment level is a more detailed description of speech than the phone level (thus, conceptually lower than the phone level). For example, the phone "T" is often made up of a silent-like segment followed by an aspiration-like segment (i.e., a short release of breath).

After words are hypothesized by POMOW, the word verifier (Wizard [McKeown - 1977]) than makes a more careful match of the words to the acoustics before another knowledge source searches for sequences of words meeting the syntactic constraints of the grammar. The system continues in a top-down mode by appending words to the ends of these sequences. The best-rated sequence of words forming a sentence in the grammar is the recognized utterance.

Hearsay-II is very sensitive to the performance of the word hypothesizer. When 33 utterances were put into three groups according to the accuracy of the word hyothesizer (about 67 utterance words per group) the performance of the system given in Figure 1.1 was found. The sentence error rate is the percent of sentences for which the system missed at least one word. The processing time does not include the time for bottom-up word hypothesization. It is clear that the performance of the word hypothesizer greatly effects the accuracy and speed of the speech system.

1.2.3 Large Vocabularies

Most of the problems of large vocabularies are typical of anything large in computer science. Investigation of these problems for a word hypothesizer are the central topics of this disertation:

> Storage requirements must be controlled and computation costs kept down as the vocabulary increases.
>
> The problem of performance degradation due to larger vocabularies must be handled. As more words are added to a

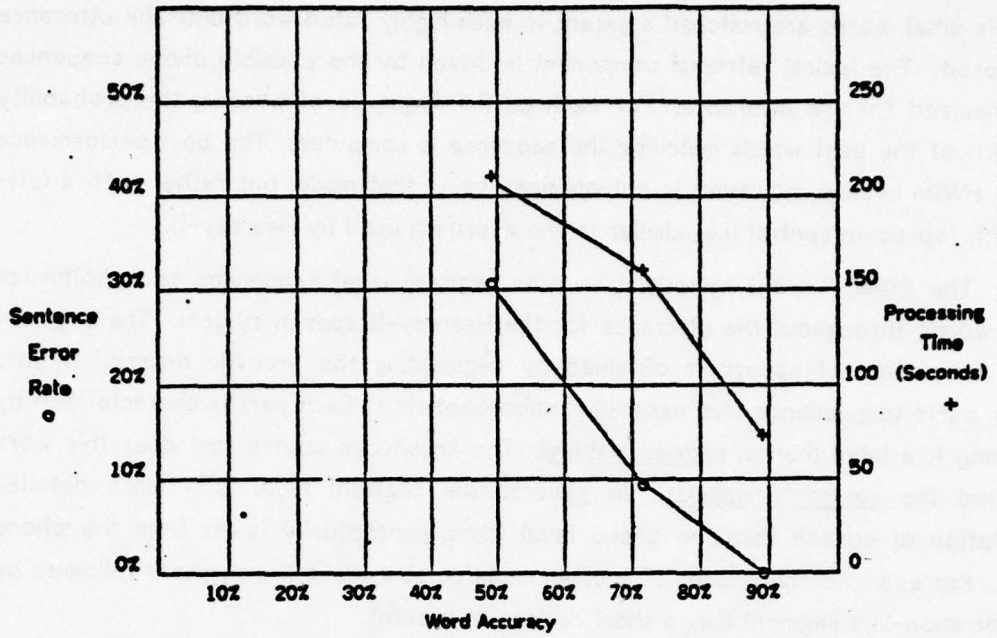


Figure 1.1: Performance of Hearsay-II versus Word Hypothesization Accuracy

vocabulary, the more likely it is that words will be confused with one another. Sometimes this is because one word is a subpart of another, such as "Plea" being confused with "Please". Other times the confusion comes because words have similar acoustic descriptions and the word hypothesizer (and/or the segmenter-labeler) cannot distinguish between the acoustic patterns. An example of this is the words "What" and "Watt".

A potentially crippling problem is knowledge acquisition. This is a common and important problem for AI knowledge-based systems [Feigenbaum - 1977]. For each word to be recognized by the word hypothesizer, a description of how it will appear in speech is needed. This acoustic description must somehow include all the variations which the word will undergo due to the problems mentioned above (coarticulation, pronunciation variations, etc.). One can acquire the acoustic descriptions for 1000 words by hand, but the amount of work required becomes increasingly prohibitive for 10,000 words or 100,000 words. Thus, a necessary goal for the Noah word hypothesizer is the ability to add words to its vocabulary easily.

Related to the problem of knowledge acquisition is the problem of flexibility. If the acoustic descriptions are tied to a particular segmenter-labeler, major effort must be spent to take advantage of an

improved segmenter-labeler. A goal for Noah is the ability to change to a new (and better) segmenter-labeler with little effort expended.

1.2.4 Human Performance

Although it is hard to constrain a human to the role of a bottom-up word hypothesizer, some measure of human performance is found in the work of Miller and Isard [Miller & Isard - 1963]. Part of the their work involved testing the ability of subjects to recognize (i.e., repeat back) the words of ungrammatical "utterances" spoken to them (e.g., "The built a was tamer fortune blaze by lazy"). Thus, the syntactic and semantic constraints were removed, forcing the subjects to recognize the words from the acoustic information alone. In a test of 50 utterances (5 to 9 words each), 56.1% of the utterances were repeated back exactly (i.e., all words recognized) and 88.3% of the principal words (i.e., not function words like "the" and "a") were recognized. The accuracy for function words was found to be lower (however, those numbers are not available). Despite a preliminary training period for the task, the subjects improved significantly during the experiment. The sentence accuracy for the first 10 sentences was 35.7%; the accuracy for the last 10 sentences was 62.1%. (Word accuracy was not given.)

Although the experiment does not indicate how much bottom-up word hypothesization is necessary for human speech understanding, it does indicate that humans can do better bottom-up word hypothesization, than machines can. Humans are able to use the acoustic information to a great degree to constrain the interpretations of speech.

1.2.5 Summary

As the constraints of vocabulary size, syntax, and semantics are relaxed for a speech understanding system, the problems of speech (e.g., coarticulation, noise, pronunciation variation, etc.) require a greater dependency by the system on the acoustic constraints present in speech. A bottom-up word hypothesizer is a component of the speech system which uses low level acoustic information (generally, phone or segment-label hypotheses) and outputs word hypotheses in order to constrain the possible interpretations of an utterance. It has been found that humans have the ability to use to a great degree (relative to current speech systems) the acoustic information alone, in order to recognize words in an utterance.

The problems of word hypothesization for a large vocabulary (and to some degree the goals and contributions of the thesis) are: 1) Knowledge Representation: storing the acoustic knowledge of words efficiently for fast retrieval; 2) Performance: hypothesizing many of correct words and few incorrect ones within "reasonable"

computation constraints; 3) Knowledge Acquisition: obtaining the acoustic knowledge for a large number of words easily; and 4) Flexibility: permitting improvements to be made to the acoustic processers of the system (e.g., segmenter-labeler) without requiring an expensive reacquisition of knowledge.

1.3 Previous Research

As has been mentioned, existent bottom-up word hypothesizers are confined to the lexical retrieval component of the HWIM system and the POMOW word hypothesizer of Hearsay-II. The structure and performance of the HWIM hypothesizer will be compared to Noah later in the thesis (see Section 6.4.2); this section is limited to a brief description of the design, performance, and limitations of POMOW.

1.3.1 Design of POMOW

POMOW introduces an intermediate level between the segment level and the words. At this new level, classes of syllables called syltypes are hypothesized based on segment-label hypotheses. Syltypes are related to the underlying sequence of segment-labels by using a Markov probability model[5]. Then, for each syltype hypothesized, all words containing a syllable which is a member of that syltype class are suggested for hypothesization. Multisyllabic words which match poorly against adjacent syltype hypotheses are pruned. We discuss below the definition of syltypes, how the Markov probability model relates them to a sequence of segment-labels, and how words are hypothesized from the syltypes.

Figure 1.2 gives a sample from a Hearsay-II word-phoneme dictionary[6]. The syntax of the dictionary permits an AND/OR tree of the possible phonemes in a word. Parentheses and commas indicate an OR group and concatenation of the elements (phonemes or syllables) indicate an AND group. Angle brackets, "<>", are syllable boundaries, with the number after the opening bracket giving the stress level of the syllable. (We use 0, 1, and 2 to indicate reduced, normal, and stressed, respectively, with a default stress of normal). A "*" in any OR group (whether composed of phonemes or syllables) indicates that the group may be absent. Many phonological rules have been put into this dictionary as alternative pronunciations. For example, "about" is defined (in Figure 1.1) to start with an optional syllable which, if present, is a reduced schwa (AX); it concludes with a stressed syllable made up of the three phonemes "B", "AW", and "T".

---

5 The idea of using the Markov probability model came from the Dragon speech system [Baker - 1975].

6 The phonemes are give in a two character Arpabet notation -- see Appendex A.

```
ABOUT        (<0AX>,*)<2B AW T>
ACUPUNCTURE <2AE K><0Y(UW,AX)><1P AX NX K><0 (T ,*)SH ER>
AGRICULTURE <2AE G><0 R (IH,IX)><1K (AX,AA) L><T SH ER>
AIRPLANE     <2(EH,EY) R><P L EY N>
AIRPLANES    <2(EH,EY) R><P L EY N Z>
AKRON        <2AE><K R (AX N ,EN)>
ALBERTA      <AE L><2B ER><(DX,T) AX>
ALCOHOL      <2AE L><0K AX><(HH,*) AA L>
ALL          <AO L>
ALLIGATOR    <2AE><0L (IH,IX)><1G EY><0 (DX,T)ER>
AMERICAN     <0(AX M ,EM)><2(M ,*)EH><0R (IH,IX)><K (IX N ,EN)>
ANALYSIS     <0AX><2N AE><0L AX><S (IH,IX) S>
AND          (<0 EN>,<(AE, IX) N (0,D)>)
ANIMALS      <2AE><N (IH,AX ,EM)><M (AX L ,EL)Z>
ANY          <1(IH,EH)><0W (IY,IH)>
ARCARO       <AA R><2K EH><R (AX,OW)>
```

Figure 1.2: Sample from a Word-Phoneme Dictionary.

The definition of syltypes is based on grouping the phonemes into seven classes: A-like vowels, I-like vowels, U-like vowels, liquids, nasals, stops, and fricatives. Figure 1.3 gives the class membership for the phonemes. Each class contains two states, depending on which side of the syllable nucleus the phoneme appears (e.g., phoneme "T" is mapped to a STOP!LEFT state if it preceeds the syllable nucleus, and to a STOP!RIGHT state if it follows the nucleus). Vowels are also mapped to left and right states. Typical state transitions are described by the network given in Figure 1.4. For example, let the above phoneme classes be represented by the symbols A,I,U,L,N,P, and F respectively (as in the first column of Figure 1.2). The word "AIRPLANES", with the pronunciation <EH R> <P L EY N Z>, is mapped into the syltypes IL and PLINF. A unique path in the network corresponds to each of these syltypes.

CODE	SYLTYPE	PHONEME
A	A-LIKE:	AE,AA,AH,AO,AX
I	I-LIKE:	IY,IH,EY,EH,IX,AY
U	U-LIKE:	OW,UH,U,UW,ER,AW,OY,EL,EM,EN
L	LIQUID:	Y,W,R,L
N	NASAL:	M,N,NX
P	STOP:	P,T,K,B,D,G,DX,
F	FRIC:	HH,F,TH,S,SH,V,DH,Z,ZH,CH,JH,WH

Figure 1.3: Phoneme Equivalence Classes.

The Markov probability model gives a way of calculating the probability of each path through the network (i.e., each syltype) for a segment-label sequence. The probabilities used in the model are inferred training utterances. A training program

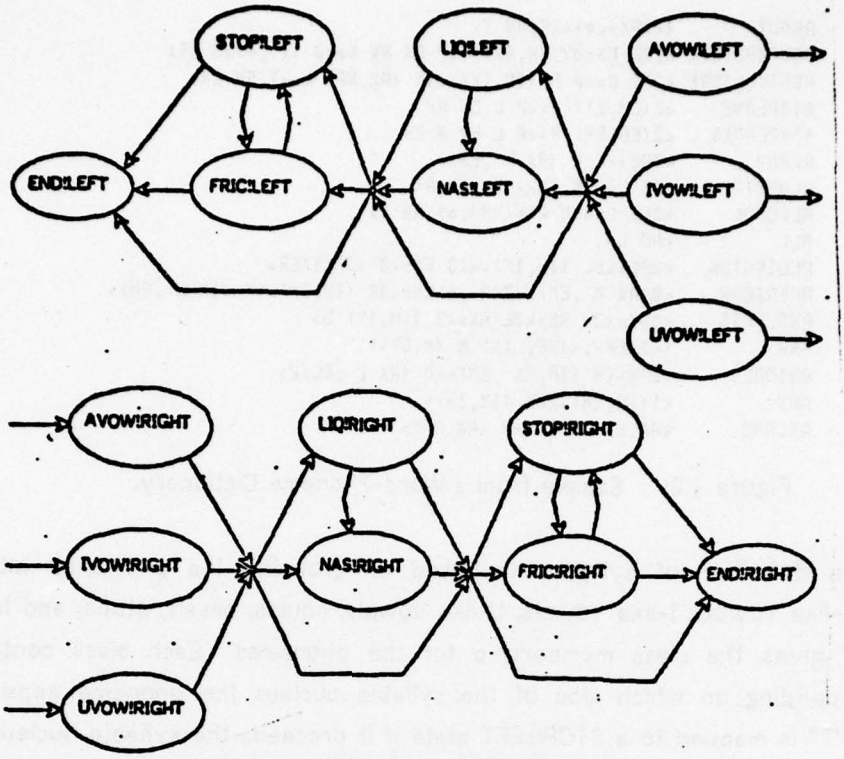


Figure 1.4: Syltype State Network

uses the phoneme equivalence classes to convert phonetically hand-labeled utterances[7] into state sequences. Previous results from the Hearsay-II segmenter-labeler are aligned with these states so that frequency counts of <current state, next state, next segment-label> triples can be made. These are normalized to give the probability of going from a current state to a new state given the next segment-label. Thus, knowledge acquisition is done by inferring probabilities from sample data.

The sequence of segment labels, and therefore the syltype state network, is not traced left to right (i.e., in the direction of increasing time) but from the syltype nucleus out to the END!LEFT state and then from the nucleus to the END!RIGHT state, as indicated by the arrows in Figure 1.4. This gives a starting point common to alternative syltypes (since all syltypes have a nucleus).

Once we have syltypes supported by the observed segment-labels, we must find the words which best fit the syltypes. The ideas behind the hypothesization of words from the syltypes are less sophisticated: Each hypothesized syltype suggests a set of words to hypothesize. A particular word is included in the set if it has a syllable which maps to the syltype and the syllable was marked in the word-phoneme dictionary as

7 That is, a person using a transcription of the utterance, the acoustic waveform, and perhaps a spectrogram makes a phonetic transcription of the utterance.

having enough stress to reliably indicate the word's presence in the utterance. (The decision of whether to mark a syllable as stressed is ad hoc.) Multisyllabic words in this set are rejected if they match poorly with adjacent syltype hypotheses. The match uses a measure based on the conditional probability that a word's syltype occurs given that a particular hypothesized syltype is observed. These probabilities are derived from a model of syltype ambiguity and the data-derived probabilities of segment-label to phoneme class confusion. Words not rejected are passed to the word verifier (Wizard [McKeown - 1977]) of the Hearsay-II system. Words rated well by the verifier are hypothesized. Thus, POMOW is used as a word filter for the word verifier.

1.3.2 Results for POMOW

The performance measure relevant to POMOW's task of filtering the vocabulary for the word verifier is the number of correct word hypotheses (i.e., hypotheses matching utterance words in name and position but not necessarily the best-rated word hypothesis for the position) and the total number of words hypothesized per utterance word. The following results are based on testing 48 utterances, none of which were included in the 60 training utterances. A 1011-word vocabulary was used. These results are for POMOW alone; the results for POMOW and Wizard as a pair will be given in Chapter 6 in comparison with Noah's results.

Performance		
Percent of words in utterance hypothesized correctly:	65%	
Avg. number of words hypothesized per utterance word:	90	
Size		
Program:	20K	(36 bits)
Permanent storage:	13K	
Word and Syltype Knowledge for 1011-Word Vocabulary:	11K	
Total:	44K	
Computation Costs		
Number of million instructions per second of speech:	9	
Times real-time for a PDP-KL10:	6.9	

1.3.3 Limitations of POMOW

POMOW's inability to discriminate between words beyond that permitted by the

definition of syltypes (equivalence classes of syllables) makes its performance and computation costs very sensitive to the size of the vocabulary. An increase in the vocabulary from 520 words to 1011 words increases the number of unique syltypes only from about 200 to 250. Thus, as the vocabulary increases, the average number of words that are supported by each syltype increases almost as fast, causing a rapid decrease in performance and speed.

Another problem effecting performance is with syllable equivalence classes defined by the syltypes. A syllable is a member of one and only one syltype class; however, in practice it is impossible to separate syllables into strict classes. The problem stems directly from the attempt to assign the phonemes to one of seven classes, as given in Figure 1.2. Phonemes which tend to lie "between" classes cause a loss in discrimination between the classes for POMOW, which results in a loss in discrimination between different syltypes. This contributes to the high figure of an average of 90 word hypotheses (which is 9% of the vocabulary) for each utterance word.

The solution for both problems is to increase the precision of the syltypes, decreasing the number of syllables per syltype class, and in the limit to use syllables rather than syltypes. This permits better performance at the cost of increasing the computation for hypothesizing syltypes or syllables at the syllable level. However, this may be more than balanced by a decrease in computation for hypothesizing words from the syllables. It is in this direction that the design of Noah was taken.

1.3.4 Conclusions for POMOW

The performance of POMOW by itself (65% correct word hypotheses competing with an average of 90 incorrect hypotheses) was found to be too low for the goals of the Hearsay-II system. For this reason, the Wizard word verifier was used to test all syntactically legal initial and final words of the utterance. This raised the effective bottom-up word accuracy for the system to above 75%. For less constrained grammars and larger vocabularies, this would not be feasible. With larger vocabularies, POMOW's performance would degrade considerably, its computation time would increase almost linearly, and its method of acquiring necessary word pronunciation variations would become unwieldy. In spite of these problems, POMOW served to clarify the problems and advantages of doing bottom-up word hypothesization and, with the use of Wizard at the ends of the utterance, it has been a key component of Hearsay-II.

1.4 Overview of Noah

The Noah word hypothesizer solves POMOW's performance, knowledge acquisition, knowledge storage, and computation problems for large vocabularies. This is done by basing word hypotheses on complete syllables rather than syltypes, separating a priori word knowledge (base pronunciations of words) from segmenter-labeler-specific knowledge, and storing the knowledge in a uniform and efficient representation.

Noah uses two levels between the input level of segment-label hypotheses[8] and the output level of word hypotheses, where POMOW uses one. These are 1) the sylpart level, consisting of parts of syllables -- onsets (the initial non-nucleus part of syllables), vowels, and codas (the final non-nucleus part of syllables) -- and 2) the syllable level, consisting of complete syllables (not syltypes as found in POMOW). Knowledge is stored in a hierarchy-tree representation. That is, between each pair of adjacent levels (segment-sylpart, sylpart-syllable, and syllable-word level pairs) is a tree structure storing a sequence of lower level units to define a higher level unit. The last node of the sequence of lower level units points to the defined higher level unit. For example, the syllable-word tree stores sequences of syllables defining each word in the vocabulary. The tree between each pair of levels permits merging common initial parts of sequences to reduce storage costs and recognition time.[9] Thus, the words "confide" and "confuse" share the first syllable node, "con", in the tree, which then points to subnodes "fide", "fuse", etc. (Section 3.2 gives an example of such a tree.)

The knowledge stored in the two higher level trees, the sylpart-syllable tree and the syllable-word tree, is obtained by processing a word-phoneme dictionary similar to the one used by POMOW. However, in this dictionary only a base pronunciation is given for each word. During processing of the dictionary some of the words also receive alternate pronunciations based on schwa deletion rules. (Schwas are reduced vowels that are sometimes deleted in normal speech. An example is the second syllable, "AX", in "summary" (S AX M - AX - R IY), which is often deleted giving S AX M - R IY.)

---

8 Segment-labels were described in Section 1.2.2 in reference to POMOW's input. A more complete description of the segmenter-labeler used by Noah (and POMOW) appears in Chapter 2. However, it is important to note that the design of Noah permits it to use speech that has been labeled but not segmented (i.e., uniformly divided into 10ms samples, for example), although at an increase of storage and computation costs. (However, this mode has not been experimented with.) This will be discussed in Section 5.2.3.1.

9 The Lexical Retrieval Component of the HWIM system also uses a tree between the phone level and the word level, for storing word pronunciations. Noah borrows this idea and expands it to a hierarchy of trees.

The knowledge stored in the segment-sylpart trees[10] is acquired by training the system on speech that has been hand-segmented into onset, vowel, and coda segment patterns. By aligning this data with the segment-labels produced for the speech by the same segmenter-labeler which will be used during recognition, segment patterns are learned for the sylparts.

Several methods are used to handle the coarticulation problems common at these levels. One method is the ability to learn a vowel-sequence, which may occur when more than one syllable (as hand-segmented) share the same syllable nucleus (as detected by the segmenter-labeler). A second is context-learning, which involves learning the surrounding segment-context of a segment-pattern in order to account for variations in the segment-patterns learned for a sylpart.

Word hypothesization in Noah is a bottom-up recognition process through four levels: 1) Syllable nuclei are recognized at the segment level; 2) Vowels, onsets, and codas are hypothesized and rated at the sylpart level, based on the segment labels; 3) syllables are hypothesized and rated at the syllable level, based on the sylpart hypotheses; and 4) words are hypothesized at the word level, based on the syllable hypotheses. The recognition algorithm between each pair of levels is very similar: A search is made through each tree based on the lower level hypotheses. Whenever such a path defines a higher level unit, that unit is hypothesized. Ratings for the lower level hypotheses direct the search and determine the rating of the higher level hypothesis.

1.5 Organization of Chapters

The chapters follow the above overview of Noah. Chapter 2 describes the hierarchy-tree representation by a simple example and as a member of a series of representation structures. Various measures of the effectiveness of this representation in reducing storage and recognition costs and for the analysis of the recognition algorithm are also given in the chapter. Since one of the measures depends on the performance of the segmenter-labeler used by Noah, a description of segmenter-labeler is also presented.

Chapter 3 describes the hierarchy-tree representation applied to Noah, showing also the storage of auxiliary information. Other representations used in speech recognition systems are also discussed.

The acquisition of knowledge is the topic of Chapter 4. The dictionary knowledge stored in the sylpart-syllable tree and the syllable-word tree and the

10 Three trees are used between the segment and sylpart levels, corresponding to the three part division of syllables into onsets, vowels, and codas.

segment-pattern knowledge stored in the segment-sylpart tree is presented.

Chapter 5 gives the major steps in the recognition algorithm and explains how the ratings are computed for the hypotheses at each level.

The performance and runtime characteristics of Noah are given in Chapter 6. This performance is analyzed to try to understand how it is effected by the vocabulary size, the amount of training, and various characteristics of the vocabulary words. This performance is compared to the performances of POMOW-Wizard and the HWIM Lexical Retrieval Component.

The final chapter includes a summary, some conclusions, and a look at future work for bottom-up word hypothesization.

1.6 Hints to the Reader

For the reader who wants to take this thesis in measured doses, the following is suggested.

> Small dose: Chapter 1 (omitting Section 1.3 on POMOW), Section 6.1, for the discussion of performance measures and test conditions, and Chapter 7.

> Medium dose: Chapter 1, first two sections of Chapter 2, first four sections of Chapter 3, Chapter 4 through Section 4.3.2 and Section 4.3.4 on context learning, Chapter 5 through Section 5.2.2, Chapter 6, and Chapter 7.

> Maximum dose: All chapters, but perhaps taking them first in the doses prescribed above.

Chapter 2: Structures and Measures

2.1 Introduction

The intent of this chapter is to present, by simple example, the structure of the knowledge representation used in Noah and to describe tools for analyzing the effectiveness of the knowledge representation. In order to evaluate the representation we will want to be able to measure the efficiency of knowledge storage, the efficiency of knowledge retrieval, and the propagation of error occurring during knowledge retrieval.

2.2 Knowledge Representation Structures

The purpose of the word hypothesizer is to convert an input of segment labels into an output of word hypotheses. The basic knowledge needed to do this is the sequence of sounds which make up each word. In particular, the hypothesizer needs to store a sequence of symbols for each word which can be compared to an input sequence of segment labels. A word is recognized (correctly or incorrectly) by a close matching[1] of the stored sequences of symbols for the word with the input sequence of segment labels.

An obvious representation for storing sequences is illustrated by the "toy" example of Figure 2.1. The defining sequence (in this case a string of lower case letters) is stored explicitly for each word.

---

1 If the symbols stored for a word represent the ideal sounds in the word (e.g., phonemes) then the matching of stored symbols to segment labels will use a distance metric between each symbol and segment label. However, if the symbols stored for each word are actually segment labels, then the match simply is a comparison of stored labels and input labels. (Input segment labels may include a rating indicating how likely the label really occured in the speech.)

```
W1 : abcde
W2 : dedfg
W3 : abcdfg      Storage cost: 25 symbols
W4 : deabc       Recognition cost:
                        21 matches per input symbol
```

Figure 2.1: Simple Storage Structure

The cost of recognition is based on a top-down recognition algorithm which 1) matches all stored sequences entirely before the closest matching word is chosen[2] and 2) does not use word boundary information - in particular the ends of the input sequence - to constrain word matching; every stored sequence is matched at every position in the input sequence. With a very large vocabulary, two necessary goals of the knowledge representation are efficient storage of the sequence for each word and efficient matching of a new input sequence with all stored sequences. This representation does not meet either goal.

One method of reducing storage is to find common subsequences in the words and then to replace each subsequence by a new symbol as is done in text file compression [Rubin-1976]. If L is the length of the subsequence and N is the number of times it appears, then the storage is reduced by N*L - (L + N) cells. When all subsequences are replaced by a new symbol, a three level hierarchy structure is formed to represent the knowledge. Figure 2.2 shows the effect of this on the example.

```
W1 : S1 S2       S1 : abc
W2 : S2 S3       S2 : de
W3 : S1 S3       S3 : dfg
W4 : S2 S1
                 Storage cost: 23 symbols
                 Recognition cost: 11 matches per input symbol
```

Figure 2.2: Hierarchy Structure[3]

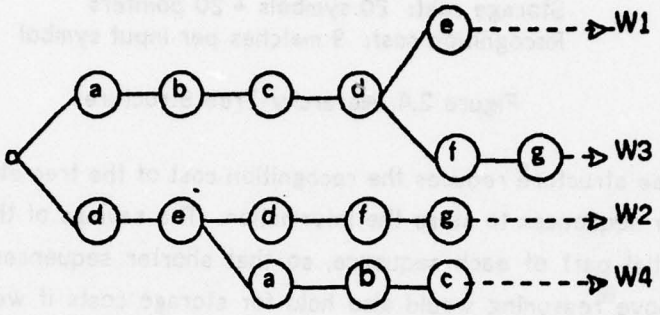
Though this structure reduces storage and recognition costs, it does so by using two assumptions which are possible weak. The first assumption is that the realization

---

2 In practice heuristics can be developed to abort the matching of a sequence based on the partial score of its initial match. Since such heuristics work as well (or perhaps better) with the other structures presented here, we will not include their use in the cost estimate. The example itself was chosen to illustrate the relative advantages of each structure when used to store knowledge for word hypothesization. Other examples can be found to favor a particular representation.

3 The recognition cost is explained in the next section.

of a subsequence is independent of the surrounding context. This is true in text compression applications where there is no uncertainty concerning the input. In speech recognition, however, we must be sure that the subsequences are context independent, or we must at least adjust for any dependency. The second assumption (which accounts for the reduction in recognition cost) is that the input sequence can be separated into regions such that each contains one subsequence (one S_i in the example). When this assumption fails, the hierarchy structure increases rather then decreases the recognition cost.



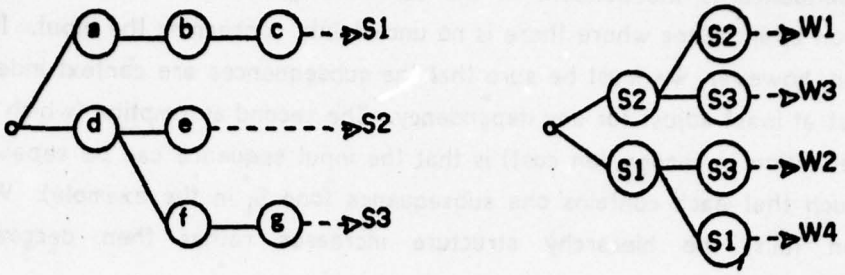
Storage cost: 19 symbols + 19 pointers
Recognition cost: 15 matches per input symbol

Figure 2.3: Tree Structure

Another method of reducing recognition and possibly storage costs is to let sequences share common initial sequences by merging all sequences into a tree structure as shown in Figure 2.3[4]. The storage of symbols has decreased but total storage may increase with the addition of pointers. Note that the terminal nodes of each path points to a "leaf" cell which contains the word defined by the path. (A nonterminal node can also point to a leaf; consider storing "W5" defined by "ded").

It is possible to combine the recognition savings of the hierarchy structure and the tree structure by forming a tree between each level of the hierarchy and creating what we will call the hierarchy-tree structure. We show an example of this structure in Figure 2.4. We can explain the decrease in recognition costs of this structure over the previous two structures in two ways. From one viewpoint, the hierarchy-tree structure reduces the recognition cost of the hierarchy structure by storing the sequences at each level in the more efficient tree structure. From another viewpoint, we can say that

4 See [Knuth-1973] Vol. 3, pp. 481 and following for a description of a "Trie" structure which is also used to store this type of information.



Storage cost: 20 symbols + 20 pointers
Recognition cost: 9 matches per input symbol

Figure 2.4: Hierarchy-Tree Structure

the hierarchy-tree structure reduces the recognition cost of the tree structure by using more but shorter sequences to store the information. The savings of the tree structure occur at the initial part of each sequence, so that shorter sequences permit greater savings. The above reasoning would also hold for storage costs if we did not include the cost of storing pointers. Chapter 3 describes how Noah uses a hierarchy-tree structure (with four levels) to represent its knowledge.

2.3 Measures of Storage and Recognition Costs

To speak precisely about different representations, we need to measure their storage and recognition costs. In particular, we will be concerned with the performance of the hierarchy-tree representation compared to the simple storage structure. This comparison is done in two steps for both storage and recognition costs. In the first step, the storage and recognition costs for each level of the hierarchy are compared to the storage and recognition costs incurred if the particular level had been omitted (i.e., if the simple storage representation had been used). In the second step, the storage and recognition costs of each tree between two adjacent levels are compared to the storage and recognition costs incurred if the tree had been omitted (i.e., if the simple storage representation had been used). In each case, ratios of the costs show the comparison. These measures will be applied to the structures in Noah in Section 3.5.

Consider a hierarchy-tree representation with D levels numbered from D (the highest level) to 1 (the lowest level). Let U_i be the number of unique symbols at level i, $N_{i,j}$ be the total number of symbols at level i needed to define the symbols at level j (i<j, i.e., the sum of the lengths of the sequences at level i), $L_{i,j}$ be the average length of the sequences of level i defining the symbols of level j, and $T_{i,j}$ be the number of

nodes used by tree between levels i and j (i=j-1) to store the sequences of symbols of level i (needed to describe the symbols at level j). The ratio of storage costs when using level i in a hierarchy to the storage costs found when level i is omitted is:

$$\text{Eq. 2.1 Hierarchy Storage: } HS_i = \frac{N_{i-1,i} \log_2 U_{i-1} + N_{i,i+1} \log_2 U_i}{N_{i-1,i+1} \log_2 U_{i-1}}$$

The ratio of recognition costs for level i of a hierarchy to the recognition costs found when level i is omitted is:

$$\text{Eq. 2.2 Hierarchy Recognition: } HR_i = \frac{N_{i-1,i} + N_{i,i+1} / L_{i-1,i}}{N_{i-1,i+1}}$$

The storage cost ratio is the cost of storing the sequences of symbols of level i-1 defining the symbols of level i (expressed in the number of bits of storage[5]), plus the cost of storing the sequences of symbols of level i defining the symbols of level i+1, divided by the cost of storing the sequences of symbols of level i-1 to define the symbols of level i+1, which would be necessary if level i was omitted.

The recognition cost ratio is more of an estimate. The number of matches per input symbol of level i-1, which are required to recognize the symbols of level i+1, are computed with level i (in the numerator) and without level i (in the denominator) to give the ratio. When using level i, two levels of matches are made. First all stored $N_{i-1,i}$ symbols of level i-1 must be matched with each input symbol to recognize symbols of level i. Then the $N_{i,i+1}$ stored symbols of level i must be matched with these newly recognized symbols of level i to recognize symbols of level i+1. However, this second cost must be adjusted to get the cost per input symbol of level i-1. This is done by dividing the second cost by the average number of level i-1 symbols for each level i symbol (i.e. the average length of the sequences at level i-1 defining the symbols of level i)[6]. For example, the recognition cost for Figure 2.2 is 8 + (8 / 2.67) = 11 symbols per input symbol.

Given two levels of the hierarchy, i-1 and i, the ratio of the storage cost found when using a tree to store the sequences of level i-1 to the storage costs when using the simple storage structure is:

$$\text{Eq. 2.3 Tree Storage: } TS_i = \frac{T_{i,i+1} (\log_2 U_i + \log_2 T_{i,i+1})}{N_{i,i+1} \log_2 U_i}$$

---

5 To be precise, the least integer greater than $\text{Log}_2 U_i$ should be used instead of $\text{Log}_2 U_i$, but since these equations will be used to compare the representation for different data bases, we will use the continuous function.

6 This falsely assumes that the sequences of level i-1 defining the symbols of level i are equally likely to occur in the input.

The ratio of recognition costs is:

Eq. 2.4 Tree Recognition: $TR_i = T_{i,i+1} / N_{i,i+1}$

The storage cost for a tree includes the cost ($\log_2 T_{i,i+1}$) of storing one pointer for each symbol.

2.4 The Confusion of Hypotheses

The recognition algorithm for the hierarchy-tree structure uses the tree at each level to convert an input sequence of units at one level into a new sequence of units at the next higher level. The recognition is complicated, however, by the nature of the lowest level input obtained from the segmenter-labeler. Since the segmenter-labeler is using only part of the information necessary for speech recognition (i.e., a localized part of the acoustics, but not the full utterance, nor syntax, semantics, prosodics, etc.), its segmentation and labeling is uncertain. It could communicate this labeling uncertainty by putting out the best label choice for each segment and letting the rest of the speech system use a precomputed label-to-label distance metric, or it could output a list of labels (as competing hypotheses) for each segment together with a corresponding list of ratings. The second method loses less information then the first and is used by the Hearsay-II segmenter-labeler [Goldberg, Reddy, and Gill - 1977]. This uncertainty in the lowest level input propagates up to higher levels of the speech system until other knowledge can constrain the possible hypotheses to hopefully the correct sequence of words and the correct semantic interpretation. In order for the recognition algorithm of Noah to handle this uncertainty, it must be able to search many paths simultaneously in each tree and store the best sequences at each level. We discuss the method of doing this in Chapter 5.

One goal of this thesis was to develop means of analyzing how the uncertainty of one level is reduced (or maybe increased) as recognition proceeds to the next higher level. One can view the tree between levels i and i+1 as a syntax filter on the uncertainty of the data at level i to obtain less uncertain data at level i+1. We need a way of measuring the uncertainty of the information in the list of symbols at each position for each level, in order to see how each step in the recognition algorithm contributes to the goal of reducing the uncertainty of the acoustic information to the correct sequence of words for the utterance.

In the remainder of this section, we first give a brief description of the segmenter-labeler in order to understand the nature of the input for Noah. We then

describe a measure for the confusion of hypotheses, which will be applied in Section 5.4 to the segment labels and the hypotheses generated by Noah in order to analyze the recognition algorithm.

2.4.1 Segmenter-Labeler Hypotheses

The following description of the segmenter-labeler used by Noah is taken from [Erman - 1977].

> Four parameters are derived by simple algorithms operating directly on the digitized audio signal (9 bit sampled at 10 KHz.) and are used by the segmenter as the basis for an acoustic segmentation and classification of the utterance. This segmentation is accomplished by an iterative refinement technique: First silence is separated from non-silence; then, the non-silence is broken down into the sonorant and non-sonorant regions, etc. Eventually, five classes of segments are produced: silence, sonorant peak, sonorant non-peak, fricative, and flap. Associated with each classified segment is its duration, absolute amplitude, and amplitude relative to its neighboring segments. The segments are contiguous and non-overlapping, with one class designation for each. (A slightly finer classification of these segments produces the segment class labels used by Noah for identification of syllable nuclei -- discussed in Section 4.3.3.1.)
>
> The labeler does a finer labeling of each segment. The labels are allophonic-like; there are currently 98 of them (see Appendix X). Each of the 98 labels is defined by a vector of auto correlation coefficients [Itakura - 1975]. These templates are generated from speaker-dependent training data that have been hand-labeled. The result of the labeling process, which matches the central portion of each segment against each of the templates using the Itakura metric, is a vector of 98 numbers; the i'th number is an estimate of the (negative log) probability that the segment represents an occurrence of the i'th allophone in the label set.

Evaluating the performance of the segmenter-labeler is difficult. This is due, in part, to the difficulty of setting a standard for comparision. One method that has been used is to compare the segment label output to a hand-made segment label description of the words in an utterance. This is done automatically by using the Harpy speech system (which uses the same segmenter-labeler) in a "forced recognition" mode as follows: the Harpy speech system recognizes an utterance by finding the path through a network of labels that matches best with the output of the segmenter-labeler (i.e., the combined rating of all the labels defined by the path is best). The network combines a description of all possible sequences of words which the grammar permits and all possible sequences of labels for each word which the word-allophone dictionary permits. By finding the best path through the sequences of labels for the correct words of an

utterance, a "correct" sequence of labels is defined. This "correct" sequence can be used to measure the distribution of the "correct" labels from the segmenter-labeler by rank or by rating. Thus, the standard for correct labels is derived from a hand-made word-allophone dictionary in which several alternate labels are given for each allophone-position in the word. ([Lowerre - 1976] gives this dictionary in an appendix.)

The following performance, based on over 26,000 segments of speech, is observed [Goldberg - 1977]:

Rank of label:	1	2	3	4	5	6	7	8
Cumulative Accuracy:	42%	58%	66%	71%	75%	77%	80%	81%

The distribution of accuracy for the rating (a value between 0 and 127) of the correct label minus the rating of the best label (which will have the lowest value of a set of labels) in groups of 10:

Rate(correct) - Rate(best):	0→9	10→19	20→29	30→39	40→49	50→127
Accuracy in group:	57%	12%	9%	6%	4%	12%

Thus, for example, 6% of the time the correct label is rated worse than the best-rated label by 30 to 39 points.

2.4.2 A Measure of the Confusion of Hypotheses

How can we measure the uncertainty of the 98 labels for a segment and for other competing hypotheses which are generated by Noah? One measure which immediately comes to mind is Shannon's entropy measure [Shannon - 1948]. The entropy measure is restricted to mutually exclusive events, but the segment labels are not viewed as mutually exclusive events by the segmenter-labeler. Rather, it attempts to measure the likelihood of each label occuring in a segment of speech and, in general, this is done independent of the likelihood that another label occurred in the same segment. In effect, the segmenter-labeler assigns pseudo probabilites as the likelihood measure for a label. A pseudo probability is a likelihood measure which has meaning on a relative scale but not on an absolute scale. For example, if hypothesis h_1 has a pseudo probability of 1.0 and hypotheses h_1 has pseudo probability of 0.5, one can say that h_1 is twice as likely of being correct than h_2. However, in general a probability of 1 does not mean that the hypothesis is "certainly" correct; nor need the probabilities of competing hypotheses sum to unity.

If we consider the pseudo probablities $(p_1,p_2,...,p_k)$ assigned to a set of competing hypotheses $(h_1,h_2,...,h_k)$ by a knowledge source (e.g., the segmenter-labeler) to be an accurate estimate of reality for a set of input conditions, then:

$$F(h_i) = \frac{\sum_{j=1,k} p_j}{p_i}$$

gives the average number of times the same conditions must hold in a series of tests in order for hypothesis h_1 to be correct once[7]. We define the number $F(h_1)$ to be a measure of the competition in a set of hypotheses for hypothesis h_1. For example, if three competing hypotheses with pseudo probabilities p_1=.9, p_2=.9, and p_3=.45 are hypothesized by a knowledge source under a set of conditions (and the probabilities are an accurate estimate of reality), then the same conditions must occur 2.5 times (=(.9+.9+.45)/.9) on the average for every time h_1 (or h_2) is correct and 5 times (=.9+.9+.45)/.45) on the average for every time h_3 is correct. Thus, the competition for h_1 (and for h_2) is 2.5; the competition for h_3 is 5. For a set of k competing hypotheses having equal pseudo probabilities, the competition for each hypothesis is k. Therefore the competition measure of an hypothesis gives the equivalent number of equally probable competing hypotheses.

We define the confusion for a set of hypotheses to be the competition in the set for the hypothesis with the greatest pseudo probability. Thus, the confusion measure for a set of hypotheses equals the competition for the best hypothesis of the set:

$$G(h_1,h_2,\ldots,h_k) = \frac{\sum_{j=1,k} p_j}{\max_i p_i}$$

The confusion measure is simply the recipocal of the normalized probablity of the best hypothesis in the set (where the probablities are normalized to sum to 1). The measure is therefore unchanged by a multiplication of the probablities of the set by a constant.

In Noah a new hypothesis at one level is formed by concatenating two or more adjacent hypotheses and giving the new hypothesis a probability equal to the product of the probablities of the old hypotheses. We now show that the confusion of a set S of hypotheses (computed by G(S)), formed from all possible pairs of hypotheses from two sets of adjacent hypotheses, T and U, equals G(T) times G(U). Let the pseudo probabilites for the two sets be $(p_1,p_2,\ldots,p_k)$ and $(q_1,q_2,\ldots,q_m)$; the probablites for the new set S will be

$$\text{Prob}(S) = (p_1q_1,\ldots,p_1q_m,p_2q_1,\ldots,p_2q_m,p_kq_1,\ldots,p_kq_m).$$

The confusion is:

$$G(S) = \frac{p_1q_1+p_1q_2+\ldots+p_kq_{m-1}+p_kq_m}{\max_{ij} p_iq_j}$$

7 Since a knowledge source may have only partial information on which to base its decisions (as with the segmenter-labeler), it is reasonable to assume that different hypotheses will be correct at different times for the same partial information.

Simplifying the sum gives:

$$G(S) = \frac{\sum_{r=1,k} p_r \sum_{s=1,m} q_s}{\max_i p_i \max_j q_j} = G(T) \times G(U)$$

The segment-label ratings generated by the segmenter-labeler can be converted to "accurate" pseudo probabilites by using a performance evaluation of the segmenter-labeler. The rating of a segment label is converted by subtracting from it the rating of the best-rated label in its segment and looking up for the result an observed accuracy in a table similar to the final table of Section 2.4.1. This accuracy value is used as the pseudo probability of the label. The pseudo probabilitites for the segment labels and for other hypotheses produced by Noah can then be used to estimate the confusion for sets of competing hypotheses. In Chapter 5, we trace the confusion of hypotheses from the input of segment labels to the output of word hypotheses for the Noah recognition algorithm.

Chapter 3: Representation of Knowledge

3.1 Introduction

The patterns describing the vocabulary words known to Noah are stored in a hierarchy-tree structure with four levels. These levels from top to bottom are the word level, the syllable level, the sylpart level (containing onsets, vowels, and codas[1]), and segments.[2] Conceptually, one tree is used to join each adjacent pair of levels; however the segment level is joined to the sylpart level by three separate trees corresponding to the three part division of the syllables into onsets, vowels, and codas.

In this chapter, we give examples of the hierarchy-tree structure applied to the knowledge in Noah (Section 3.2), attempt to justify the levels (Section 3.3), and show what auxiliary information is stored in the representation (Section 3.4). Measures of storage and recognition costs, developed in Chapter 2, are applied in Section 3.5 and the actually storage costs are given in Section 3.6. Finally in Section 3.7 we compare this representation to others being used in speech recognition.

3.2 An Example

Consider the sample dictionary of Figure 3.1. Each word is followed by its pronunciation given in two character ARPABET Computer phonetic representation[3], with hyphens indicating syllable boundaries. The second syllable of "ACM", for example, is made up of the vowel "IY", the onset "S", and a null coda. Parentheses enclose a list of options separated by a comma with a "*" representing a null option. In this sample dictionary, the first syllable of "about" can be dropped optionaly.

---

1 An onset is the initial nonnucleus part of a syllable; a coda is the final nonnucleus part.

2 We will use the term "segment" to mean a labeled segment of speech, and "segment label" to refer to one of possibly many labels assigned to a segment.

3 See Appendix A for the correspondence between the ARPABET representation and the International Phonetic Alphabet.

A	EY
ABOUT	(AX- ,*) B AW T
ABSTRACT	AE B - S T R AE K T
ABSTRACTION	AE B - S T R AE K - SH IX N
ABSTRACTS	AE B - S T R AE K T S
ACL	EY - S IY - EH L
ACM	EY - S IY - EH M

Figure 3.1: Sample Dictionary

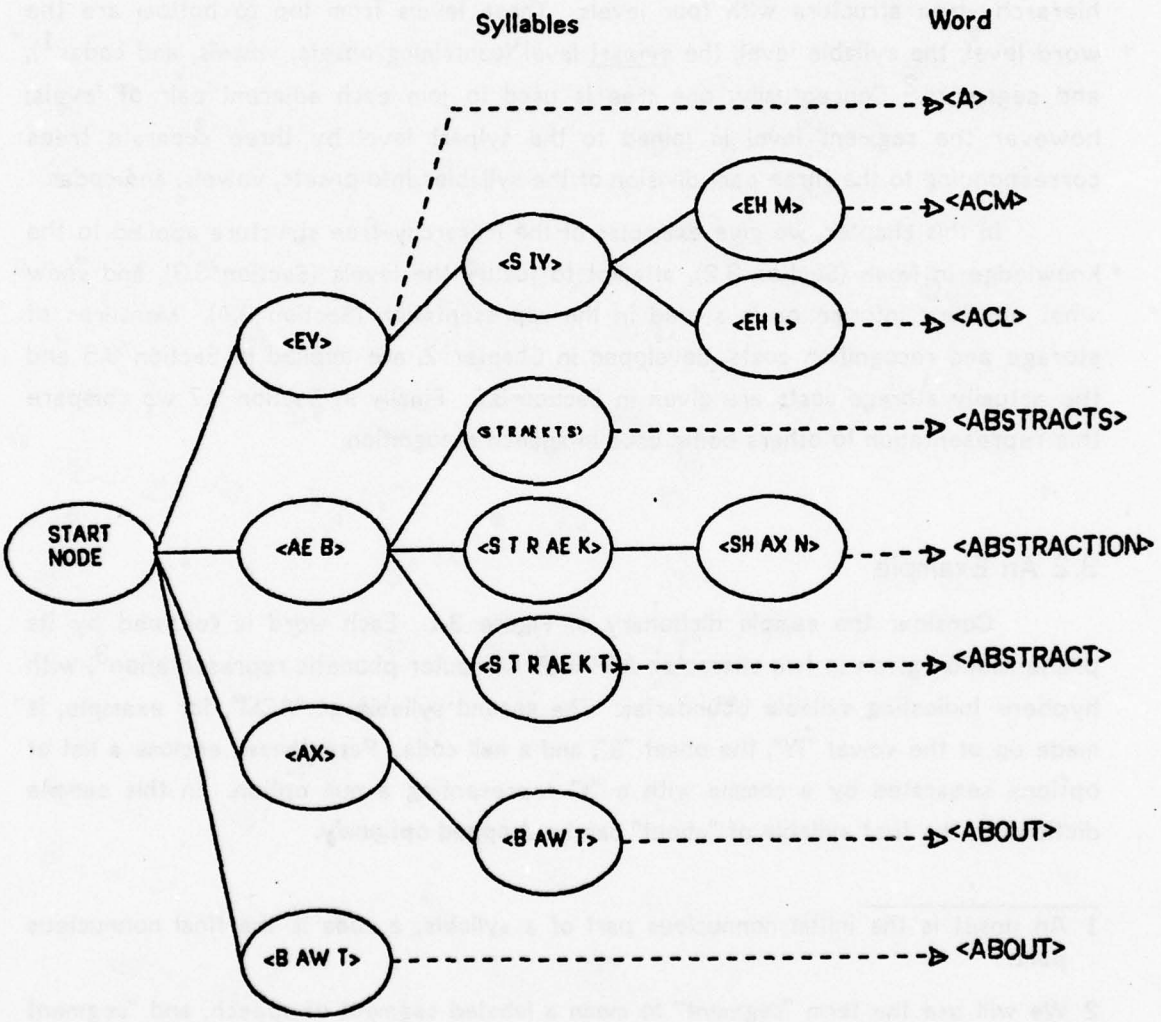


Figure 3.2: Syllable-Word Tree for Sample Dictionary

The syllable-word tree (i.e., the tree joining the syllable level to the word level) for this small dictionary is shown in Figure 3.2. In this figure solid lines indicate pointers between nodes of the tree[4] and dashed lines indicate pointers to leaves. Associated with each level is a lexicon of items used to describe speech at that level. For example, the syllable level has a corresponding lexicon of syllables. One of the items in this lexicon is the syllable "B AW T". (The angle brackets in the figure indicate a unique lexicon number for the enclosed quantity; the syllable itself is not stored at a node, but rather its lexicon number.) It is possible for one path of nodes to point to more than one leaf if words share the same pronunciation (homonyms). Also, it is possible for the same word to appear on separate leaves if it has distinct pronunciations (like "About" in this sample dictionary).

Figure 3.3 gives the corresponding sylpart-to-syllable tree. The main thing to notice about this tree is that the sylparts of the paths do not follow the left-to-right order found in a syllable. The vowels are put first to simplify recognition (which will be discussed in Chapter 5). Each path in this tree has three non-terminal nodes: a vowel, an onset, and a coda. The symbol "*" in the figure represents a null onset or coda.

So for we have represented only the knowledge found in the dictionary -- what about the knowledge which characterizes the output of the segmenter-labeler? The mapping between the idealized speech given by the pronunciations of the dictionary and the actual speech represented by the labeled segments is accomplished by the segment-sylpart trees. Figure 3.4 shows the segment-label patterns learned for some of the codas present in the sample dictionary. A segment-label pattern is the sequence of segment labels produced by the segmenter-labeler for a particular sylpart (particular codas in this case)[5]. Each coda is followed by a list of alternate segment-label patterns. Segment labels are enclosed in brackets to distinguish them from the phonemes in the dictionary. For example, the coda "K T S" has three possible segment patterns, the first of which is a sequence of three segment labels [-] (silence), [K], and [S]. The tree storing these patterns is given in Figure 3.5. The tree gives a many-to-many mapping of the segment patterns onto the codas. The segment-vowel tree and segment-onset tree are similar; however the segment-onset tree stores the patterns in reverse order (i.e., right-to-left) and the segment-vowel tree stores them in an order depending on the pattern (to be explained Section 4.3.3.2). It is in these trees, between the segment level and the sylpart level, that most of the ambiguity found in speech is stored for this

4 The pointers shown are conceptual. The trees are implemented in a binary tree representation (see [Knuth - 1968], Vol. 1, pp. 238), i.e., the sons of any node are elements on a linked list.

5 Section 4.3 describes how these patterns are learned.

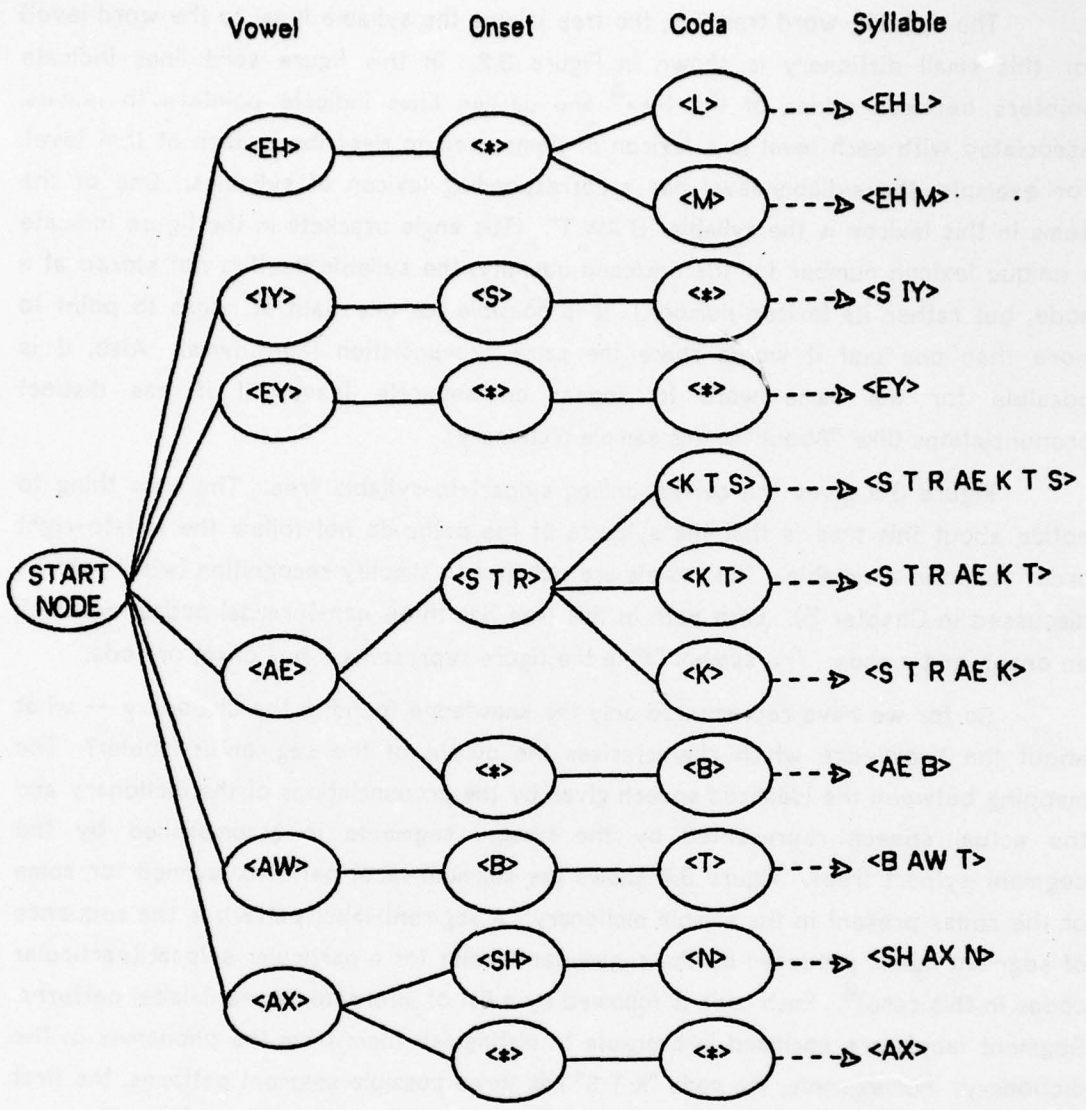


Figure 3.3: Sylpart-Syllable Tree for Sample Dictionary

Coda	Segment-label patterns
T	[-], [DX], [←], [← B], [- T]
B	[B], [←]
KT	[- T], [← D], [- S],
KTS	[- K S], [- S SH], [- S K]

Figure 3.4: Sample segment patterns for Codas

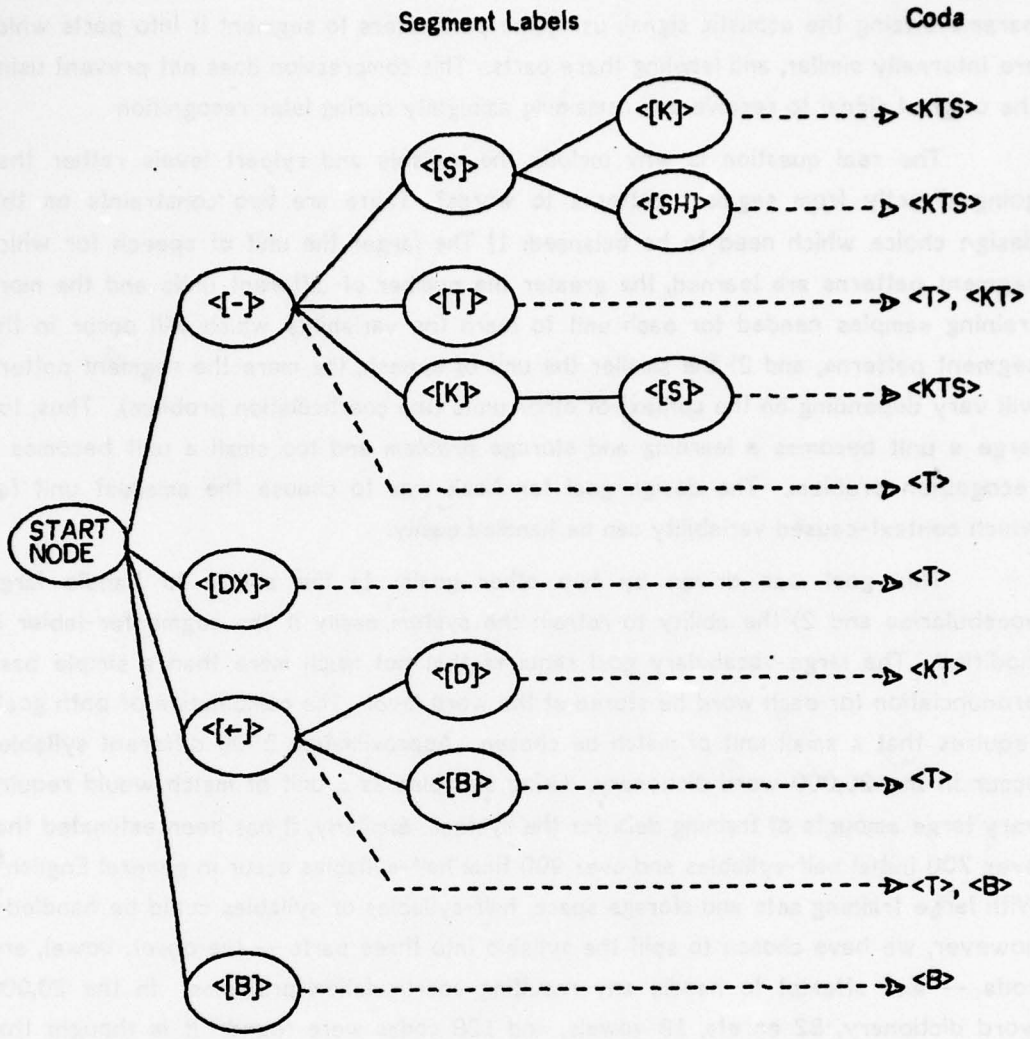


Figure 3.5: Sample Segment-Coda Tree

system. Each sylpart has different segment patterns representing it, and each segment pattern may represent more than one sylpart. The next section discusses why these levels were chosen for Noah.

3.3 Levels of Speech Representation

The reasons for chosing a word level and a segment level are clear. Words are needed because we are interested in word hypothesization and are the ultimate outputs of the hypothesizer. Segments compress the speech information to make pattern

recognition tractable. The information of the speech wave is compressed by parameterizing the acoustic signal, using the parameters to segment it into parts which are internally similar, and labeling these parts. This compression does not prevent using the original signal to resolve any remaining ambiguity during later recognition.

The real question is why include the syllable and sylpart levels rather than going directly from segment patterns to words? There are two constraints on this design choice which need to be balanced: 1) The larger the unit of speech for which segment patterns are learned, the greater the number of different units and the more training samples needed for each unit to learn the variability which will occur in the segment patterns, and 2) the smaller the unit of speech, the more the segment pattern will vary depending on the context of other units (the coarticulation problem). Thus, too large a unit becomes a learning and storage problem and too small a unit becomes a recognition problem. The design goal for Noah was to choose the smallest unit for which context-caused variability can be handled easily.

This goal was driven by two other goals: 1) the ability to handle large vocabularies and 2) the ability to retrain the system easily if the segmenter-labler is modified. The large-vocabulary goal requires that not much more than a simple base pronunciation for each word be stored at the word level. The combination of both goals requires that a small unit of match be chosen. Approximately 5900 different syllables occur in the 20,000-word dictionary. Using syllables as a unit of match would require very large amounts of training data for the system. Similarly, it has been estimated that over 700 initial half-syllables and over 900 final half-syllables occur in general English[6]. With large training sets and storage space, half-syllables or syllables could be handled[7], however, we have chosen to split the syllable into three parts -- the onset, vowel, and coda -- and attempt to handle any resulting coarticulation problems. In the 20,000 word dictionary, 82 onsets, 18 vowels, and 128 codas were found. It is thought that with this size of unit of match, the proper balance between recognition performance versus storage and training sample size is found for a large vocabulary word hypothesizer.

A still smaller unit of speech for matching with the segments could have been the phoneme. Coarticulation problems make it difficult for the phoneme to be recognized accurately from the segment labels. Experience with a "phone synthesizer" knowledge source in Hearsay-II [Shockey & Adam - 1976] made this clear. Even when

---

6 See [Sivertson - 1961] for a summary of estimates on the numbers of different sized units in speech.

7 Half-syllables, syllables or maybe even words may have to be used to obtain the detail necessary for word verification systems.

phonemes are recognized bottom-up in speech, several word pronunciations must be stored for each word in order to account for coarticulation effects. This is done in the HWIM system by storing about six pronunciations per word.

There are two methods which Noah uses to handle coarticulation problems inherent in using onsets, vowels, and codas. The first is called "context learning" and is used with all three sylparts; the second is called "vowel sequence learning" and is used as needed for vowels. The first method is mentioned in the next section but both are described fully in chapters 4 and 5 on learning and recognition.

Once the sylpart level is chosen, the syllable level becomes a natural bridge between the sylpart level and word level for efficient recognition in the hierarchy-tree structure. It will be shown in Section 3.6 that including the syllable level in the hierarchy-tree structure for a 12,000-word vocabulary reduces the estimated recognition costs to less than one-fourth the estimated cost incurred without the level, at a slight increase in storage costs.

3.4 Auxiliary Information

In addition to the knowledge represented in trees as described above, context information is stored at the leaves of the segment-sylpart trees. Associated with each sylpart pointed to by each segment pattern stored in the segment-sylpart trees (i.e., with each leaf pointed to by each path in a tree), is a list of segment-label pairs. Each pair gives the left and right context segment-labels of the segment pattern occuring when the pattern was learned for the particular sylpart. For example, Figure 3.5 shows three sample lists of segment-label context pairs for the coda "T" and three for the coda "B". The first context segment-labels learned for the segment pattern [DX] of coda "T" are "R" on the left and "IH3" on the right. One can detect some dependencies between the patterns and contexts shown in the figure. Generally, the coda "T" appears as a [DX] in the context of two vowels (the top-left column), but as the pattern [← B] in the context of a vowel and a nasal or a liquid (the top-right column). Also, if the pattern [←] is produced by the segmenter and labeler in the context of a vowel and a fricative, it is more likely that the coda "B" was spoken rather than the coda "T" (the center two columns). It is these kinds of dependencies which permit the word hypothesizer to constrain the interpretations of a particular segment pattern. This will be described in Chapter 5.

Also stored with each sylpart (i.e., each leaf) in the segment-sylpart tree is a count of the number of times the segment pattern (represented by the path in the tree pointing to the leaf) has appeared for the particular sylpart. This information is used to compute a weight penalty, as described in Chapter 5.

Coda: T

Pattern: [DX] Context		Pattern: [←] Context		Pattern: [← B] Context	
Left	Right	Left	Right	Left	Right
R	IH3	IH3	OW3	AA5	W
AA5	ER	IH	NX	AYX	M1
AA5	IH3	ER2	-	EYR	EL
ER2	AYR	EYC	IH2	ER2	W
UH4	AE	IY	NX		
EH4	AE2	AYR	D		

Coda: B

Pattern: [B] Context		Pattern: [←] Context		Pattern: [-] Context	
Left	Right	Left	Right	Left	Right
AA3	S	AA5	S	AE3	S
AE3	S	AYC	ZH	OW2	S
AO	EL3	AYR	S	AYR	S
AO	IH6	AA4	EL2	IH	-
ER	S	AYC	SH		

Figure 3.6: Sample Segment-Label Context for Codas T and B

3.5 Application of Storage and Recognition Measures

Of what value are the various parts of the hierarchy-tree structure for storage and recognition efficiency in Noah? In an attempt to answer this question we gathered statistics for the parts of the structure and applied Equations 2.1 through 2.4. The knowledge acquired by Noah is divided into dictionary knowledge, obtained from a word-phoneme dictionary and stored in the sylpart-syllable and syllable-word trees, and segment-label knowledge, acquired from segmented and labeled training utterances and stored in the segment-sylpart trees (this is explained in Chapter 4). Because of this division of knowledge, the statistics and the results for the parts of the hierarchy-tree structure are separated into two groups -- one using the knowledge obtained from the dictionary and one using the knowledge obtained from the training utterances. Table 3.1A shows the statistics for the dictionary knowledge for three sizes of vocabularies; Table 3.1B shows them for the segment-label knowledge for 174 training utterances.

The measures of storage and recognition cost ratios for the hierarchy part and the tree part of the hierarchy-tree structure are computed using these numbers and the equations of Section 2.3. For example, to compute the ratio of the estimated recognition cost of using a syllable level to the estimated recognition cost of not using it for the 1000-word vocabulary we obtain from Eq. 2.2:

Words in Vocabulary:	1011	4020	12049
Unique Syllables:	1012	2669	4584
Syllables to define Words:	2304	10253	31375
Nodes in Syllable-Word Tree:	1784	7469	20281
Unique Sylparts:	151	151	151
Sylparts to define Syllables:	2684	7439	12971
Sylparts to define Words:	5426	23618	71987
Nodes in Sylpart-Syllable Tree:	1406	3286	5333

Table 3.1A: Statistics for 1000-, 4000-, and 12,000-Word Vocabularies.

Total Words in Training:	1106
Total Syllable Nuclei:	1653
Unique Sylparts:	118
Total Sylparts:	3950
Sylparts to define Syllables:	1261
Unique Segment Labels:	98
Total Segments:	7038
Segments to define Sylparts:	3038
Nodes in Segment-Sylpart Trees:	1716

Table 3.1B: Statistics for 174 Training Utterances

$$HR_{Syllable} = \frac{N_{Sylpart,Syllable} + N_{Syllable,Word} / L_{Sylpart,Syllable}}{N_{Sylpart,Word}}$$

From Table 3.1A we see that $N_{sylpart,syllable} = 2684$, $N_{syllable,word} = 2304$, $L_{sylpart,syllable}$ (the average length of syllables in sylparts) = 2684/1012 = 2.7, and $N_{sylpart,word} = 5426$. Thus, the recognition cost ratio for the syllable level is about 0.65, which means that the estimated reduction in cost of the recognition algorithm due to the inclusion of syllable level in the hierarchy-tree structure is about one-third. Table 3.2 shows the complete results for the application of the storage and recognition cost ratios. The results are separated (by a dashed line) into two groups corresponding to dictionary knowledge (above the line) and segment-label knowledge (below the line).

We can make several observations at this point. First, the values derived from the dictionaries tend to improve as the vocabulary gets larger. This is expected; as more words, syllables and sylparts are stored, the chances of common subpatterns increase. Second, the storage cost for a tree used between two levels is always greater than a simple storage structure. This is because of the need to store pointers

Level	Measure	Vocabulary:	1000	4000	12,000
Syllable					
	Hierarchy Storage (Eq. 2.1):		1.08	1.00	0.91
	Hierarchy Recognition (Eq. 2.2):		0.65	0.47	0.34
	Tree Storage (Eq. 2.3):		1.60	1.56	1.41
	Tree Recognition (Eq. 2.4):		0.77	0.73	0.65
Sylpart					
	Tree Storage:		1.28	1.15	1.11
	Tree Recognition:		0.52	0.44	0.41
		174 Training Utterances			
	Hierarchy Storage:		0.77		
	Hierarchy Recognition:		0.67		
Segment					
	Tree Storage		1.48		
	Tree Recognition:		0.56		

Table 3.2: Results for Storage and Recognition Cost Measures

for the tree. However, we think the savings in recognition costs are worth it. (These savings are increased when various heuristics for pruning the tree search are used.) The final observation is that we can obtain a total cost ratio estimate for including a particular level and storing the units of the level in a tree by multipling the value of the storage (or recognition) hierarchy cost ratio for the level by the value of the storage (or recognition) tree cost ratio for the level. For example, the total storage cost ratio for including a syllable level for the 12,000-word vocabulary is 1.28 (= .91 * 1.41); the total recognition cost ratio is 0.22 (= .34 * .65). Thus, the estimated recognition costs are reduced to less than one-fourth by the use of the syllable level, at a slight increase in storage.

3.6 Storage Costs

Figure 3.7 shows the storage costs for the different vocabularies tested. The center curve, which includes the storage for the sylpart-syllable tree and for the non-terminal nodes of the syllable-word tree (i.e., just nodes storing the syllables of the words), shows a gradual decline in the amount of storage required for each vocabulary word. This decline shows the characteristic of the trees to share common information. The storage added onto this curve to give the top curve is almost linear with vocabulary size. This storage is due to the terminal nodes of the syllable-word tree (each of which points to a word) and various information stored for each word, such as

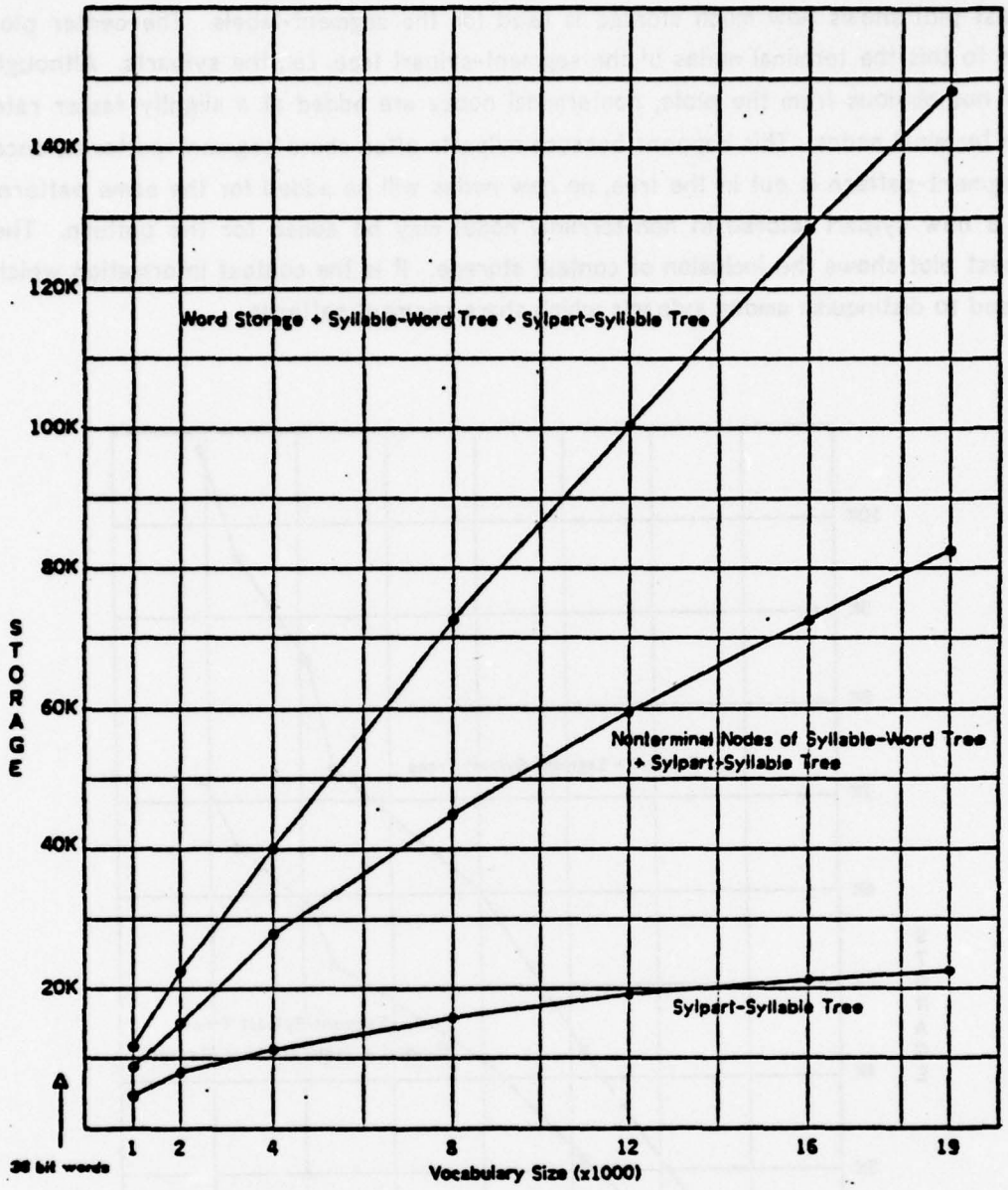


Figure 3.7: Storage Costs versus Vocabulary Size for Dictionary Knowledge

its spelling. (Word spellings use about 27K (19% of 144K) for the 19,000-word vocabulary, and are only used for analysis output.)

In Figure 3.8, the x-axis gives the number of segments for new sylpart segment-patterns for various numbers of training utterances. Rather than plot the storage costs of the segment-label knowledge versus the total number of segments in the training utterances, we ignored those segment-patterns which were redundant (i.e., had already been learned) and counted the number of segments in new segment-patterns. The

lowest plot shows how much storage is used for the segment-labels. The center plot adds to this the terminal nodes of the segment-sylpart tree, i.e., the sylparts. Although it is not obvious from the plots, nonterminal nodes are added at a slightly faster rate than terminal nodes. This happens because sylparts often share segment-patterns; once a segment-pattern is put in the tree, no new nodes will be added for the same pattern, but a new sylpart (stored in non-terminal node) may be added for the pattern. The highest plot shows the inclusion of context storage. It is the context information which is used to distinquish among sylparts which share segment patterns.

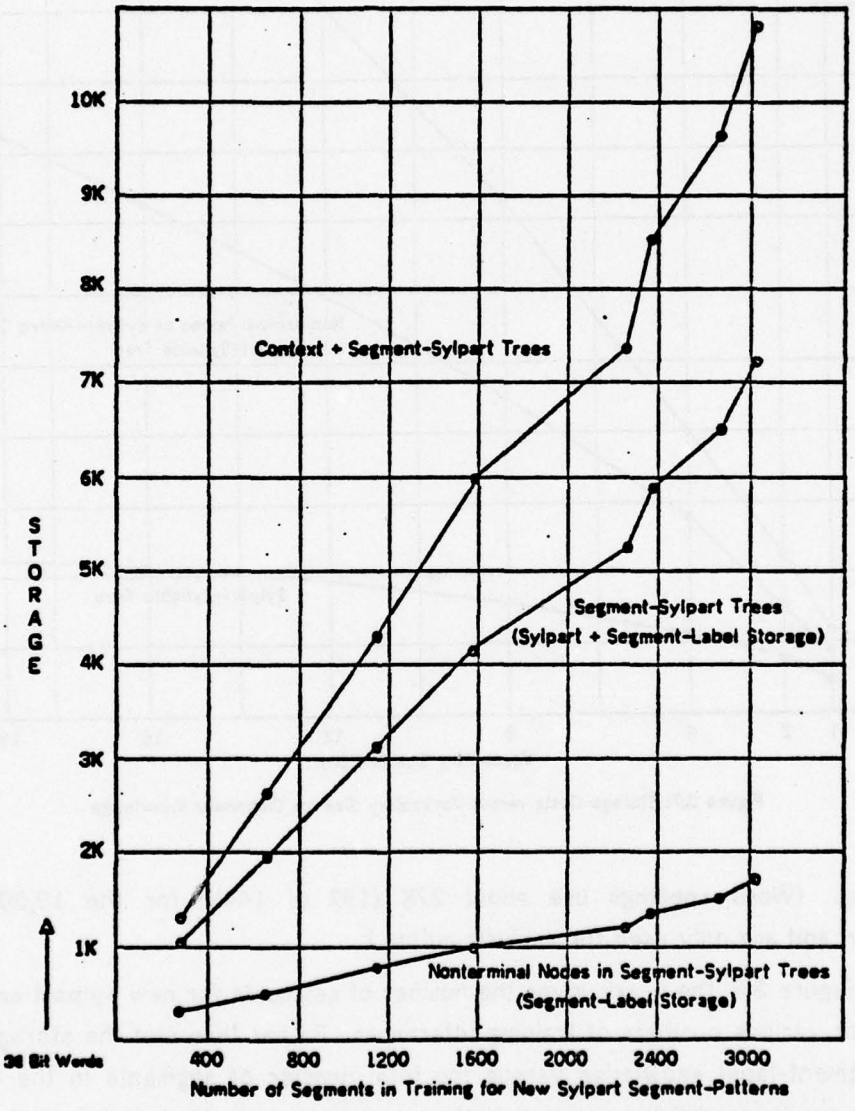


Figure 3.8: Storage Costs versus Number of Segments in Training, for Segment-Label Knowledge

3.7 Other Knowledge Representations for Speech

Many types of knowledge representations have been used or suggested for use in speech recognition. We will briefly look at the characteristics of four here, each of which is used to store sequential information. In particular, we are interested in how each might work for storing knowledge for a word hypothesizer. The four representations are 1) A tree representation used by the lexical retrieval component of the HWIM system [Klovstad - 1976], 2) A network used by the Harpy system [Lowerre - 1976], 3) Wood's Augmented Transition Network (ATN) used by the syntax-semantic parser of the HWIM System [Woods, et al. - 1976], and 4) An Automatically COmpilable Recognition Network (ACORN) [Hayes-Roth & Mowstow - 1975] used initially as the parser of the Hearsay-II system.

Comparison of these four at the level of storage and recognition efficiency is difficult, if not impossible, since each representation is used within a different framework and each for a different goal. One dimension which they can be compared is the relative amounts of knowledge content and knowledge structure each stores. The simple storage structure, for example, stores only knowledge content; the structure of the knowledge is implicit in the assumption that we are storing sequences. At the other end of the scale is the ACORN representation. In this lattice type representation each primative knowledge unit is stored only once. However, pointers and higher level nodes combine to structure the content into the same sequential information. Figure 3.9 orders several representations on this dimension.

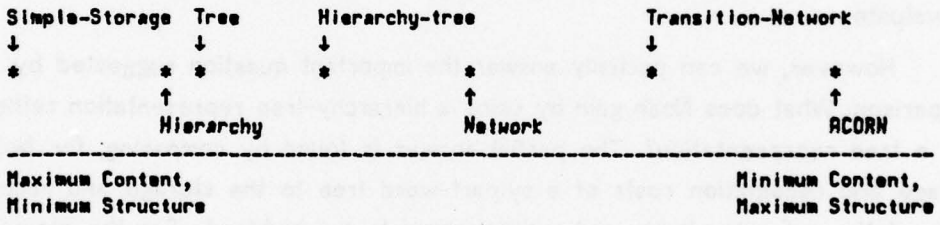


Figure 3.9: Ordering of Representations.

The ordering is by our intuition (somewhat based on experience) about how these representations would store word speech knowledge.

3.6.1 Tree

The Lexical retrieval component of the HWIM system joins a phonetic level with the word level by storing word pronunciations in a tree structure. Each word (counting inflected forms as different words) has on the average more that six pronunciations stored to account for 1) within-word variations due to palatalization, syllabification,

vowel reduction, and other phonological phenomena, 2) within-word variations due to the peculiarities of the acoustic-phonetic recognition component and 3) end-of-word variations due to the affect of preceding and following words. The latter variations, which account for two-thirds of the extra pronunciations, are constrained during word recognition if the word context is known.

It is interesting to compute the storage and recognition cost ratios for the knowledge stored in this tree structure for the HWIM system. A tree of 4371 nodes is required to store 1960 pronunciations consisting of 10616 phonemes (71 different phonemes) [Klovstad - 1976]. Using Eq 2.3 we have:

$$TS = \frac{4371*(\log_2 71 + \log_2 4371)}{10616 \log_2 71} = 1.22$$

for the tree storage ratio, and from Eq. 2.4 we have:

$$TR = 4371 / 10616 = 0.41$$

for the tree recognition ratio. Thus, for a 22% increase in storage over the simple storage structure, the tree reduces recognition costs to 41% of the simple recognition costs[8].

It is tempting to compare these ratios to the ratios of Table 3.2 for Noah, but differences in the data stored (i.e., phonemes instead of sylparts and six pronunciations per word instead of approximately one per word) make such a direct comparison hard to evaluate.

However, we can partially answer the important question suggested by such a comparison: What does Noah gain by using a hierarchy-tree representation rather than just a tree representation? The partial answer is found by comparing, for Noah, the storage and recognition costs of a sylpart-word tree to the storage and recognition costs of the sylpart-syllable and syllable-word trees combined. For the storage cost comparison, the numerator of Eq. 2.3 gives the storage cost of each tree; the ratio of the sum of the storage costs for the sylpart-syllable tree and the syllable-word tree to the storage costs of a sylpart-word tree gives the comparison:

Storage Ratio =

$$\frac{T_{Sylpart,Syllable}(\log_2 U_{Sylpart} + T_{Sylpart,Syllable}) + T_{Syllable,Word}(\log_2 U_{Syllable} + T_{Syllable,Word})}{T_{Sylpart,Word}(\log_2 U_{Sylpart} + T_{Sylpart,Word})}$$

8 Of course, these are only estimates, based on an assumed storage implementation and recognition algorithm (as described in Chapter 2). In particular, the ratio falsely assumes the ability to used 6.15 bits (=$\log_2 71$) for storing each pointer of the tree.

It was found that 8471 nodes were required in a sylpart-word tree for a 4000 word vocabulary. Using this value and values obtained from Table 3.1A we have:

$$\text{Storage Ratio} = \frac{3286(\log_2 151 + \log_2 3286) + 7469(\log_2 2669 + \log_2 7469)}{8471(\log_2 151 + \log_2 8471)} = 1.40$$

The recognition cost ratio comparison is simply:

$$\text{Recognition Ratio} = \frac{T_{\text{Sylpart,Syllable}} + T_{\text{Syllable,Word}} / L_{\text{Sylpart,Syllable}}}{T_{\text{Sylpart,Word}}}$$

$$= \frac{3286 + 7469/2.3}{8471} = 0.77$$

Thus, a 23% speed up is obtained by adding 40% more storage. Both of these values would improve for larger vocabularies.

The lexical retrieval component of HWIM also stores auxiliary information in order to constrain the search of the tree when searching for words of a particular length or words of a particular syntactic-semantic category. For example, the syntax and semantics component of the system can ask the lexical retrieval component to find all verbs matching well between two time periods in the utterance. A tree structure also exists for doing a backward search in the utterance. In this tree, the phonemes of the final parts of word pronunciation share common nodes. This permits finding all words which match the phones of an utterance occuring before a particular part of the utterance. Thus, the system can query the lexical retrieval component for all words ending at a particular time in the utterance.

3.6.2 Network

A tree combines common initial parts of sequences, but a network combines all possible parts of sequences within the constraint of preserving the uniqueness of each sequence. A tree "remembers" the past sequence whereas a network "forgets" what symbols have been traced on a particular path. It is this "forgetting" which makes a network less suitable for word hypothesization. The Harpy Speech system uses a network representation to store all possible segment-label sequences which could appear for all possible sentences generated by its grammar. Since the system needs to find only the best path through this network, it is economical to trace forward through the network to follow the best paths and then to backtrace to find the best one. A word hypothesizer might have to do an expensive backtrace to find the best N paths if it used this representation.

3.6.3 Transition Network

Harpy's recognition network is made by merging a grammar network (in which each node is a word) with the phonetic networks describing the possible patterns of each word. Before the merge is done we have a structure similar to a two level transition network. The difference is that the nodes of the grammar network are expanded to the lower level networks rather than the transitions between the nodes. In either case these structures are related to networks in the same way the hierarchy structure is related to the simple storage structure. The levels of the more complex structures permit the sharing of similar parts of the more simple structures. Whether or not the more complex structures reduce storage space or recognition time depends on the knowledge being stored. In the case of Harpy's network it may be that fewer nodes and pointers are required (for the grammars and vocabularies involved) when the two levels are merged into one network and reduced by removing null states and redundant states and by doing "subsumption" of common states[9].

An augmented transition network has been use as the parser in the HWIM speech system. The term "augmented" refers to associating an action or a set of actions with an arc between two states of the network[Woods - 1970]. These actions are to be executed whenever a path traced by the parser uses the arc. Actions are used to build up a semantic interpretation of the parse, to store current conditions of the parse, or to check various conditions necessary for continuing the parse. An arc can be one of five types for the HWIM system, corresponding to the type of action stored on it. The five types are: PUSH, POP, WRD, CAT and JUMP. The PUSH arc represents a nonterminal of the grammar and signals the parser to enter a lower level network describing the nonterminal. The POP arc signals the return from the lower level network to the original level. The WRD and CAT arcs match words and syntactic-semantic catagories of words, respectively. Thus, it is these arcs which match the words found by the lexical retrieval component. The JUMP arc permits the parser to jump between certain states without "consuming" any of the utterance. For example, two states having an "adjective" arc (which would be a CAT type of arc) between them could also have a JUMP arc between them indicating that the adjective is optional between those states according to the grammar.

The disadvantage of a transition network representation for word hypothesization is similar to the disadvantage of a network representation. A transition network is oriented towards finding the best path (i.e., best parse, in the above example) and not many best paths as would be needed for a word hypothesizer.

---

9 From conversation with Lowerre. See [Lowerre 1976] for details of this network reduction.

However, it might be possible to use an augmented transition network whose "actions" explicitly save the best word hypotheses.

3.6.4 ACORN

At the extreme structure end of the content-structure dimension we have the recognition network called an ACORN (for Automatically COmpiled Recognition Network) [Hayes-roth & Mostow - 1975], which has been suggested for speech and vision recognition and for a time served as the base of the syntactic parser of the Hearsay-II speech system. As a parser, the recognition network was automatically compiled from a description of the grammar with the goal of "maximally exploiting repeated subparts of the grammar" - i.e., the number of nodes in the network was minimized. Thus we have the case of minimizing content storage without regard to structural storage cost.

As a parser of the Hearsay-II system, the ACORN representation had unique terminal nodes corresponding to each word in the vocabulary. Nonterminal nodes corresponded to phrases (i.e., sequences of words). In this lattice type representation, a node X is linked to any and all (conceptually) higher nodes which contain the word or phrase associated with X as a subpart (a "conjunctive" grouping at the higher node) or as a member of a class of phrases or words associated with the higher node (a "disjunctive" grouping at the higher node). For example, there is a unique terminal node associated with the word "Me". This node is linked to nonterminal nodes such as those representing "Give-Me" and "Tell-Me" as part of a conjunctive grouping. The node might also be linked to a nonterminal node such as "Personal-Pronoun" which has a disjunctive grouping. (E.g., "Us and "I" might also be linked disjunctively to "Personal-Pronoun".)

During recognition, a word hypothesis triggers the unique terminal node corresponding to it. This constitutes a match of the node with the speech input. This node passes the positional and rating information of the word along its links to higher nonterminal nodes. A nonterminal node with a disjunctive grouping of lower nodes passes the information on up the network, but a node with a conjunctive grouping of lower nodes is only partially matched until all of its lower nodes "report" with words or phrases which 1) form an adjacent sequence and 2) pass a combined rating test. When these conditions are met, the node passes the new information along its links to yet higher nodes. This process continues until some top level node signals that a best complete parse has been found.

Ideally this recognition process gives a bottom-up, non-backtracking parser capable of starting its work at any and all places in the utterance. However, in practice, a combinatorial explosion occurs due to 1) the errorful nature of word hypotheses (not all of the correct hypotheses are made and many incorrect ones are made) and 2) the

number of partially matched nodes made for each match of a node at a lower level. For example, the word "Me" may be hypothesized many times incorrectly. Each time it is hypothesized, all nodes in the representation containing "Me" as a subpart are partially matched. Since not all correct word hypotheses are present, partially matched nodes must be used to predict the missing words. Thus, "Me" will be used to predict "Give" and "Tell" (from the example above). The number of partially matched nodes and the number of resulting predictions swamp the system [Hayes-Roth, Mostow and Fox - 1977][10].

It seems that a word hypothesizer using errorful segment-label hypotheses would have a similar combinatorial explosion using an ACORN representation.

10 Part of the solution to this problem for Hearsay-II was first to pass the word hypotheses through a simple parser that found sequences of words which were pairwise grammatically adjacent. The ACORN algorithm took each sequence, verified that it was grammatical (or that there was a usable subpart of it that was grammatical) and continued searching the network. Essentially, the first pass parser permitted the ACORN algorithm to begin at higher level nodes in the representation.

Chapter 4: Acquisition of Knowledge

4.1 Introduction

In this chapter we address the important issue of acquiring knowledge for the Noah word hypothesizer. Knowledge acquistion is a central problem for AI knowledge-based systems [Feigenbaum-77]. Difficulty in acquiring knowledge can prevent a knowledge-based system (no matter how ingenious) from advancing from "toy" problems to real-world problems. Special attention was given to the design of Noah so that it could acquire the knowledge necessary for a "real-world" vocabulary.

Let us consider three possible methods of acquiring knowledge. In general, knowledge-based systems can acquire their knowledge by 1) the manual method in which someone looks at the problem confronting the system and puts in the needed knowledge, usually at a very detailed level; 2) the rule method: in which the required knowledge has been reduced to a set of rules which are encoded for the system; and 3) the automatic learning method in which the knowledge from training samples. A system might use more than one method. Below we discuss how these methods appear for speech systems.

In the manual method, one looks at examples of the input in order to hand-tune word templates (or whatever the unit of the pattern match is chosen to be). After the initial tuning, this method involves a cycle of attempted recognition on new input, investigation of errors, and readjustment of templates. The method is potentially very accurate. However, it is time consuming work, it may lack generality, and it does not permit changes to the segmenter-labeler without starting from scratch. The Harpy speech system [Lowerre-1976] uses primarily this method.

In the rule method, much of the speech knowledge is encoded in the form of phonetic rules which attempt to account for the differences between the ideal word pronunciations and observed speech. These rules can be applied either in a top-down mode by modifying the dictionary word pronunciations (as is done in the HWIM speech system) or in a bottom-up mode by modifying the output of the segmenter-labeler. (An early version of Hearsay-II attempted this [CMU Computer Science Speech Group -

1976].) The problem with using the rules in a bottom-up mode is that the rules are based on a generative model of speech which describes the transformations from words to the speech signal. Unfortunately these transformations are not easily or uniquely reversible. The main problem for either the bottom-up or top-down modes is knowing when to apply the rules. "Every rule has its exceptions" is very true of linguistic rules. Variables, such as intonation and rate of speech, often modify them.

The automatic learning method requires 1) a framework which is able to contain the necessary speech knowledge and 2) preclassified samples used to put the knowledge into the framework, i.e., to train the system. The difficulties are designing the framework (knowing what type of knowledge will be needed) and getting sufficient amounts of correctly classified training data. If the framework is too general, the sample size will be too small; if it is too narrow, it may never contain enough knowledge to recognize words accurately.

Noah has aspects of all three acquisition methods. The manual method is used to acquire base pronunciations of words. This knowledge is dictionary knowledge; it is constant for a given vocabulary; it gives the patterns of words in terms of sylparts; and it is considered a priori speech knowledge. The rule method is used to modify some of the base pronunciations to account for schwa deletions (discussed below). The automatic learning method is used to acquire the patterns for sylparts in terms of segment label patterns This knowledge characterizes the particular segmenter-labeler used by the word hypothesizer and is considered learned speech knowledge. We next describe the acquisition of dictionary knowledge along with the application of the schwa deletion rules. This is followed by a description of the acquisition of segment label knowledge (Section 4.3).

4.2 Dictionary Knowledge

One goal for Noah was the ability to acquire and store the knowledge for a very large number of words easily. This is made possible by using a standard pronunciation dictionary as the source of knowledge. A computer-readable 20,000-word pronunciation dictionary[1]. Parts of this dictionary (every Nth word) were combined with the 1000-word Hearsay-II dictionary (containing the words of the test sentences) to obtain various sizes of test vocabularies.

Only one type of pronunciation variation was added to the base pronunciations found in the dictionary. Variations due to vowel deletions are added by rule as the

1 We used the pronunciations from "The New Meriam Webstr Pocket Dictionary - 1964" received from Richard Goldhor and Jon Allen at MIT as produced by John Olney and Donald Ramsey (described in [Olney & Ramsey - 1972]).

dictionary is processed[2]. For example, the second schwa in the word "summary" (S AX M - AX - R IY) is commonly deleted so that the word is often pronounced S AX M - R IY. Since the schwa is not in the speech, the only way that a bottom-up syllable-based word hypothesizer can recognize the word is to store the two syllable pronunciation.

Knowing when to delete a schwa is not easy. Cole, who has summarized and indexed the phonological rules of the ARPA speech community [Cole - 1974], concludes his summary of 16 vowel deletion rules with the statement: "it seems that the process of vowel deletion is a rather complex one, not yet well understood". One problem is that the deletion rules, summarized by Cole, go beyond reasonable schwa deletions as found in the words "summary", "reference" (R EH F - (AX -) R AX N S), and "cardinal" (K AA R D - (AX -) N EL), to words which must be considered exceptions to the rules such as "agony" (AE G - (AX -) N IY) and "element" (EH L - (AX -) M AX N T). A second problem is that schwa deletion depends on the rate and manner of speech. For example, deleting the schwa in the word "karate", giving (K R AA T - IY), might occur in fast speech, but in carefully articulated speech it would be considered mispronounced. Such is not the case with the word "summary".

```
Sample Rule 1: Delete AX in left context of <stressed syllable, final B>,
       and right context of <unstressed syllable, initial R>.
              Elaborate        1H - 1L AE B - (AX -) R AX T
              Laboratory       1L AE B - (AX -) R AX - 2T OW - IY
              Robbery          1R AA B - (AX -) R IY

Sample Rule 2: Delete AX in left context of <stressed syllable, final K>,
       and right context of <unstressed syllable, initial L>.
              Broccoli         1B R AA K - (AX -) L IY
              Chocolate        1T SH AA K - (AX -) L AX T
              Stickler         1S T IH K - (AX -) L ER

Sample Rule 3: Delete AX in left context of <stressed syllable> and initial D,
       and right context of <unstressed syllable, initial R.
       Merge D with R.
              Boundary         1B AW N - D (AX -) R IY
              Mandarin         1M AE N - D (AX -) R AX N


Examples of words in which schwas were not deleted:
              Academy          AX - 1K AE D - AX - M IY
              Widower          1W IY D - AX - W ER
              Liberation       2L IH B - AX - 1R EY - SH AX N
              Illiterate       IH L - 1L IH T - AX - R AX T
              Ritual           1R IH T SH - AX - W EL
              Vocalize         1V OY - K AX - 2L AY Z
              Parliament       1P AA R - L AX - M AX N T
```

Figure 4.1: Sample Schwa Deletion Rules and Examples.

2 The pronunciation format of the words, as described in Chapter 2, permits specifying any type of pronunciation variation. However, this feature was not used.

In deciding what schwas to delete, we chose to limit the deletions to what might occur in carefully articulated speech. Using the rules collected by Cole as a reference we studied[3] the contexts in which schwas occured in the 20,000-word dictionary. This resulted in the set of rules given in Appendix C. Figure 4.1 gives three examples of the schwa deletion rules with examples of their application. In addition, the figure gives examples of words from the dictionary which contain schwas which were not deleted.

As the pronunciation of each word is read from the dictionary and added to the syllable-word tree, alternate pronunciations are added according to the schwa deletion rules. These rules expand the number of pronunciations by only 2.5%. Since these rules do not apply often to the dictionary and since the test utterances were made up of carefully articulated speech, the rules have little effect on the performance of Noah. The rules applied to only 10 instances of the 1160 words in the training utterances (discussed in the next section), and to only 3 instances of the 705 words in the test utterances. Although the schwa deletion rules could be removed with little effect, it was not clear beforehand what their effect would be. The more common occurrance of merging a schwa with a nearby vowel (so that schwa and the vowel share a syllable nuclei) is handled by vowel sequence learning, described in Section 4.3.3.3.

4.3 Segment-Label Knowledge

4.3.1 Hand Segmentation

Segment patterns for sylparts are obtained by hand-segmenting speech into sylpart-sized sections[4]. It should be noted that hand-segmentation of speech into sylpart-sized sections is not person independent. Occasionally people will disagree about where a sylpart boundary occurs or whether a "vowel sequence" (mentioned

---

3 It should be noted that that very little real speech was looked at (only the 174 training utterances) in refining the rules. The decision of whether to delete the schwa in a certain context was based on the subjective test of whether the words having a schwa in that context "sounded" right without the schwa.

4 This potentially time consuming work was simplified by using an interactive program which displays the speech waveform for each utterance on a graphics terminal and shows the segmentation, best label choice, and segment class labels of the current segmenter-labeler on the waveform. The program then queries the user word-by-word for the correct begin and end times of each sylpart. The output of the program is a file which, when used with the output of a segmenter-labeler, defines the segment patterns of each sylpart.

below) should be formed. However, the important thing for pattern recognition is consistency in hand-segmentation. Since one person (the author) did all of the hand-segmentation, some degree of consistency was obtained.

4.3.2 Segment Pattern Learning

Learning a segment pattern for a sylpart involves storing the sequence of top-rated segment labels in the appropriate segment-sylpart tree (onset, vowel, or coda)[5]. This simple scheme is modified by two methods of merging similar patterns in order to save storage space. Figure 4.2 shows the top five segment-labels for the utterance "Please Help Me". The number in parentheses before each label is the rating of the label, generated by the labeler and normalized by subtracting off the best rating; the lower the number the better the rating. The class-label is a broad characterization of the segment. Its use to Noah will be described later (Section 4.3.3.1).

SEGMENT NO.	TIME	SEGMENT CLASS-LABEL	TOP FIVE SEGMENT-LABELS AND RATINGS 1	2	3	4	5
1	(48:64)	SIL	(0) -	(16) TH	(25) F	(30) ←	(37) Z
2	(64:70)	AF2	(0) PL	(7) T	(14) S	(15) TH	(18) -
3	(70:79)	VIY	(0) IH6	(2) IH4	(7) IH7	(9) IH3	(14) IH2
4	(79:89)	NGL	(0) Y	(12) ER3	(15) G	(17) NX	(19) IX
5	(89:93)	FRV	(0) S	(19) TH	(19) PL	(20) ZH	(25) T
6	(93:98)	HUF	(0) S	(27) ZH	(36) SH	(40) PL	(45) TH
7	(98:101)	HUF	(0) T	(5) D	(6) TH	(7) K	(7) F
8	(101:107)	HHY	(0) HH	(12) V	(13) DH	(16) IH7	(17) P
9	(107:114)	VLW	(0) AWC	(1) AYL	(7) AO	(13) AA3	(18) AA5
10	(114:119)	TCN	(0) EL2	(9) EL3	(11) EL	(20) L	(23) L2
11	(119:130)	SIL	(0) -	(19) TH	(25) F	(26) ←	(38) Z
12	(130:135)	NAS	(0) M1	(6) M	(9) UW	(10) EM	(21) AA4
13	(135:144)	VIY	(0) IH5	(3) IH2	(6) IH3	(6) AYR	(7) IH
14	(144:153)	HHY	(0) IY	(4) Y	(5) IH5	(7) IH7	(8) IH3
15	(153:160)	LVF	(0) G	(1) D	(1) HH2	(6) TH	(7) DH

Figure 4.2: Partial Segment-Label Output for "Please help me".

Consider the third sylpart of the utterance, the coda "Z" of the syllable of "Please", between the horizontal lines: these lines represent the times given by the hand-segmentation. The times correspond to a top-rated segment label pattern of "[S S T]" (segments 5 through 7) for the coda "Z". However, the pattern is stored as "[S T]", which, during recognition, will match any number of S's followed by any number of T's.

5 At one time, segment duration information was learned as part of the pattern. This was discontinued when it was found that the duration information did not improve the recognition results. It seems that segment duration information is very dependent on stress, rate of speech, intonation, and other factors above the sylpart level which we were not prepared to include.

This compression of identical adjacent labels is the first method of pattern merging. The second is to use a non-top choice label if 1) its rating is below a set threshold and 2) the label is the best choice (according to the label ratings of the current segment) of all previously learned patterns which match the new segment pattern up to the current segment. This is done as follows: a pattern is added to the tree segment by segment, each segment label becoming a node. Before a segment label is added, all following nodes are checked to find the best matching segment-label (previously learned) to the current segment. If the rating of the best one is below (i.e., better than) the set threshold, its node is taken as part of the current pattern. If none of the labels has a good enough rating, or there are no next nodes, the top-rated label of the current segment is inserted as a new node. For example, suppose the segment pattern "[IH4]" has been learned for the vowel "IY", but the pattern "[IH6]" has not been learned. The learning program would interpret the segment pattern for the vowel "IY" of "Please" (segment number 3) as having already been learned, if the threshold is greater than 2 (which is the rating of "[IH4]" in segment 3). The only change in the system would be in the count of the number of times the "[IH4]" pattern had been seen for the vowel "IY".

4.3.3 Segment Pattern Learning for Vowels

In addition to the above features, there are three others for vowels: syllable nuclei, nonsequential pattern storage, and vowel sequences. To discuss these features we will refer to Figure 4.3, showing the top segment-label choices for the utterance "Do any papers cite Nilsson?" divided into sylpart sections defined by the hand-segmentation.

4.3.3.1 Syllable Nuclei

As a syllable-based recognition method, Noah needs to find where the syllables are. This is done by locating syllable nuclei. It was found experimentally that the segment class labels generated by the segmenter-labeler provided a fairly reliable indication of syllable nuclei. Segment class labels are a by-product of the segmenter: In order for the segmenter to divide the acoustic description of speech into similar parts, it must analyize and characterize the speech (usually in 10ms samples). A segment class label is simply this characterization for a complete segment. (Appendix B gives a list of the segment class labels). Any sequence of (one or more) class labels whose names begin with a "V" (which stands for "vowel") implies a syllable nucleus. For example, the class label of segment number 11 indicates the syllable nucleus of the first syllable of the word "Papers".

4.3.3.2 Nonsequential Pattern Storage

Since the syllable nucleus gives a common starting point for matching the vowel

Segment No.	Time Cent-sec.	Segment Class-Label	Top Segment-Label	Hand-Segmentation Sylpart	Code	Times
1	(51:63)	SIL	-	D Onset		(61:64)
2	(63:64)	ASH	PH			
3	(64:69)	NAS	UW	UW Vowel	V	(64:73)
4	(69:76)	VCN	ER2			
5	(76:79)	VFT	EH4	EH Vowel	V	(73:79)
6	(79:82)	DCN	M1	N Coda		(79:82)
7	(82:88)	VFT	IY	IY Vowel		(82:88)
8	(88:98)	SIL	-	P Onset		(88:100)
9	(98:100)	ASP	P			
10	(100:104)	TCN	EYL	EY Vowel		(100:112)
11	(104:109)	VFT	EYC			
12	(109:112)	TCN	IY3			
13	(112:119)	SIL	-	P Onset		(112:120)
14	(119:120)	ASP	V			
15	(120:126)	VMD	ER	ER Vowel		(120:129)
16	(126:129)	HHY	IH2			
17	(129:137)	HUF	S	Z Coda		(129:137)
18	(137:141)	HUF	S	S Onset		(137:141)
19	(141:152)	VAA	AYL	AY Vowel		(141:154)
20	(152:155)	DCN	IH2			
21	(155:161)	LVF	NX	T Coda		(154:158)
22	(161:165)	NGL	NX	N Onset		(158:165)
23	(165:174)	TCN	IH3	IH Vowel	C	(165:174)
24	(174:183)	VBK	EL3	L Coda		(174:183)
25	(183:189)	HUF	S	S Onset		(183:195)
26	(189:195)	HUF	S			
27	(195:198)	VMD	AH	AX Vowel		(195:201)
28	(198:202)	VCN	L2			
29	(202:206)	LVF	NX	N Coda		(201:213)
30	(206:213)	LVF	-			

Figure 4.3: Top Segment Labels and Hand-Segmentation for "Do any papers cite Nilsson?"

segment patterns, Noah learns the patterns beginning with the syllable nucleus. Consider again the first vowel of "papers", "EY", which spans segments 10 through 12. Segment 11 is put first in the tree, then segment 10 (and then earlier segments if they are part of the vowel), and finally segment 12 (and any later segments). Thus, the pattern will appear in the segment-vowel tree as a path through the nodes marked "EYC", "EYL", "IY3".

4.3.3.3 Vowel Sequence Learning

Often vowels run together in speech, sharing the same syllable nucleus. At other times the segment pattern of a vowel is greatly modified by the preceding onset or the following coda. Sometimes this modification of the vowel pattern is the only clue to identity of the onset or coda. It is for these reasons that a method of "vowel sequence learning" was developed. Basically, the method permits forming a "vowel sequence" by concatenating a vowel with another vowel, with the last phoneme of an onset, or with the first phoneme of a coda whenever the speech waveform gives no evidence of distinct sylparts. The segment pattern spanned by this vowel sequence is learned[6]. For example, the first two vowels of "Do any" (segments 3, 4, and 5 in Figure 4.3) share the same syllable nucleus (segments 4 and 5)[7]. The hand-segmentation code of "V" (for Vowel concatenation) in the "Code" column after each vowel indicates that these vowels should be joined and learned as a vowel sequence. The label UW-EH is added to the vowel sequence lexicon (if it is not already there) and the segment pattern from segments 3 through 5 is added to the segment-vowel tree. Recognition of this pattern identifies the vowel sequence UW-EH, which is expanded and treated as if two separate vowels had been recognized.

Another example is found in segments 23 and 24. Here the "L" of "Nilsson" has become a syllabic /L/ with the vowel appearing as a "tail consonant" (TCN) before it. The "C" (for Coda concatenation) in the hand segmentation tells the learning program to form the vowel sequence IH-L and learn segments 23 and 24 as its pattern. More complex vowel sequences occur. A common one is IY-R-IY as found in the word "Theory". In hand-segmenting 1350 syllable nuclei, 235 (17%) were found to be vowel sequences. These included 94 different vowel sequences, one-third of which were liquid-vowel or vowel-liquid pairs. (Appendix B lists the vowel sequences found for 174 training utterances).

Vowel sequence learning is the first method for handling coarticulation problems at the sylpart level; the second is context learning.

---

6 During recognition, a match of the segment pattern with the new segments identifies the same vowel sequence, which is than expanded into its parts.

7 As mentioned above, any sequence of "V" class segments defines a nucleus.

4.3.4 Context Learning

When learning the segment pattern for a sylpart, selected context about the pattern is also learned. Though this context could be of several types (e.g., sylparts to the left and right in utterance, stress of surrounding syllables, relative amplitude of the sylpart, or position of the sylpart in the utterance -- to account for end of utterance effects[8]), we have chosen to learn the one-segment pattern to the left and to the right of the sylpart's segment pattern. These adjacent segments of a segment pattern give the most relevant information about why the segment pattern for the sylpart appears as it does. Other factors, such as the types of contexts suggested above, influence the pattern, but the greatest influence is given by the speech immediately before and after the pattern. During recognition this context is used to limit the possible interpretations of a segment pattern.

In Figure 4.3, the context for the first vowel of "papers" (segments 10, 11 and 12) are segments 9 and 13. Just as in learning sylpart segment patterns, the top segment labels of segments 9 and 13 are not necessarily chosen for the context. If some other labels had already been learned for the same sylpart and segment pattern, and both labels have ratings (in their respective segments) higher than a set threshold,[9] no new context will be learned. Rather, the count of the number of times the old context appeared will be incremented.

4.3.5 Hand-made Segment Patterns

The manual method of knowledge acquisition was used for some of the less frequently occuring onsets and codas. Of the 83 onsets and 131 codas occurring in the 20,000-word dictionary, 24 onsets and 58 codas did not occur at all in our training and test sentences (i.e., in the sentences made from the 1000-word dictionary). Since these onsets and codas make up only 1.4% and 1.2% of all onset and coda occurences, respectively, in the 20,000-word dictionary, we choose to eliminate them from the lexicons. Any word using one of these sylparts is not used in any of the test vocabularies. Of the remaining onsets and codas, 14 onsets and 26 codas did not occur, or occured infrequently, in the training sentences. For these sylparts, we entered hand-made patterns into the segment-sylpart trees. A hand-made pattern for a sylpart was made by combining and modifying the patterns stored for similar sylparts. For example, the pattern for the onset "S P L" was made to be "[S - PL]". "[S]" is the most

---

8 We have experimented some with the last two, but these contexts are not used currently.

9 Since the identity of a sylpart depends less on the context of its segment pattern than on the segment pattern, this threshold is less restrictive than for the threshold used for the segment labels in the segment patterns.

common pattern for onset "S" and "[- PL]" is the most common pattern for the onset "P L". (Appendix B lists the sylparts used by Noah).

Chapter 5: Recognition

5.1 Introduction

Recognition is a bottom-up process through four levels for the Noah word hypothesizer: 1) Syllable nuclei are recognized at the segment level; 2) Vowels, onsets, and codas are hypothesized and rated at the sylpart level, based on segment labels; 3) syllables are hypothesized and rated at the syllable level, based on the sylpart hypotheses; and 4) words are hypothesized at the word level, based on the syllable hypotheses. This chapter describes the major steps in this recognition process and explains how the ratings are computed for the hypotheses at each level.

Before the recognition algorithm is discussed, it is necessary to distinguish between "lexical items" and "hypotheses": Each level has a lexicon associated with it. For example, the syllable level has a lexicon of syllables made up of all syllables which occur in any vocabulary word. The sylpart level has a lexicon, which for convenience, has been divided into the onset, vowel, and coda lexicons. A lexical item is one entry in a lexicon. In contrast, a hypothesis is a suggested realization of a lexical item in the speech utterance. It has a beginning and ending time (or segment number for this system), a lexical index which points to the lexical item it represents and a rating, measuring the likelihood that the lexical item occurs at the given place in the utterance.

5.2 The Recognition Algorithm

5.2.1 Information Needed for Recognition

Hypothesis X at level i is based on a sequence of adjacent hypotheses at the next lower level, level i-1, as explained in Chapter 2. Three types of information are combined to produce hypothesis X: 1) The sequence of lexical items at level i-1 that define hypothesis X, 2) the hypotheses at level i-1 that have been produced previously, and 3) the adjacency information of hypotheses at level i-1. Each of these types of information will now be described.

The sequence information is acquired during training of the system and is stored

in trees between adjacent levels. Chapter 4 discussed the acquisition of this information and Chapter 3 gave examples of its storage in trees.

Information about what hypotheses exist at level i-1 is generated and stored as recognition proceeds up through each level. Figure 5.1 shows sample hypotheses at each of the four levels for the last syllable in the utterance "I'd like to see the menus". The time in centi-seconds measured from the beginning of the utterance is shown for each segment on the bottom line. Immediately above the time is the sequential number of the segment. Displayed with each hypothesis is its lexical name, its rating in parentheses (lower numbers indicate better ratings) and, except for the segment hypotheses, its beginning and ending segments (indicated by a time line). Although only the top-five segment labels are shown, the segmenter-labeler produces the ratings for each of the 98 labels in the segment lexicon for each segment.

Three things should be noted concerning the hypotheses at the sylpart level: 1) One type of sylpart (vowel, onset, or coda) can overlap with other types. In other words, a syllable region is not divided into regions of onsets, vowels, and codas; the position of a sylpart is based on the match of its stored segment pattern with the segment label hypotheses. 2) A null onset hypothesis (with a duration of zero) exists at each segment position in which a vowel hypothesis begins and a null coda hypothesis exists at each segment position in which a vowel hypothesis ends. None of these hypotheses is displayed here. 3) The hypotheses "Y UW" and "IYAX" are examples of "vowel sequences". They will be discussed later (Section 5.2.3.2).

Two hypotheses at the same level are adjacent if one hypotheses ends at segment number N and the other begins at segment number N+1. This makes storing adjacency information simple -- a list pointing to all hypotheses of the same level beginning at segment number N+1 gives the set of all hypotheses adjacent to any hypothesis of that level ending at segment number N. (These adjacency lists are not necessary at the segment level, where adjacency information is implicit in the storage method or at the word level were adjacency information is not needed by the word hypothesizer). This simple definition of adjacency requires a new hypothesis for every place the pattern of a lexical item matches the lower level. For example, the vowel sequence "Y UW" appears four times, spanning different segments. However, though the alternative method of storing begin-time and end-time "fuzziness" for each hypothesis saves storage it does not permit rating each time span separately, and it needs sophisticated adjacency tests between hypotheses. (This is the method used by the Hearsay II system).

5.2.2 One Step in Recognition

The sequences of adjacent hypotheses at level i-1 are compared to the

```
                ----------

WORD LEVEL      ...MENUS(0).....................................................>
                   <.........ME(12).........>
                            <.............USE(19).....................>
                            <.............AN(23).............>
                   <.................NEWS(33).....................>

                ----------

SYLLABLE        <.........M IY(12).........>
 LEVEL                    <.............Y UW Z(19).....................>
                          <............AX N(23).............>
                          <..........IH N(25).........>
                <............M Y UW(27)............>
                <.................N Y UW Z(33).....................>

                ----------

SYLLABLE-PART      Onsets          Vowels          Codas
   LEVEL                 <.....IH(13).....> <Y(15)>
                         <.........Y UW(13).....> <....Z(12)...>
                         <Y(13)> <..Y UW(17)...> <....N(26)....>
                <M(13)> <....IYAX(16).....> <N(23)>
                <N(13)> <....Y UW(18).....> <.....V Z(26).....>
                         <DH(29)><....Y UW(18).....>  <....S(23)...>
                <...M Y (17)..> <.....UW(21).....>  <...K S(25)..>

                ----------

        Top five  NX(0)   Y(0)    IY(0)   UW3(0)  M1(0)   D(0)    S(0)
           labels: M(8)    IY(11)  Y(13)   Y(2)    UW(1)   Z(12)   SH(47)
SEGMENT           N(8)    IH5(29) EYC(17) ER3(2)  M(5)    DH(16)  ZH(50)
 LEVEL            M1(10)  G(32)   IY2(24) IH2(4)  IH7(5)  TH(17)  T(50)
                  EM(10)  IH3(34) ER3(26) IH5(7)  EM(6)   V(18)   PH(53)
        Segment
           class: NAS     FRV     VSW     VCN     NVF     LUF     FRU

                ----------

        Segment #:  20      21      22      23      24      25      26
Time (centi-secs)   117     125     130     133     139     146     152:157
```

Figure 5.1: Sample Hypotheses at Each Level
for the Last Syllable of "MENUS"

sequences of lexical items represented by the nodes of the tree between levels i-1 and level i to produce the hypotheses at level i. Assume that the sequence of m ($m \geq 1$) hypotheses $[h_1,h_2,...,h_m]$ at level i-1 have been matched with m nodes $[n_1,n_2,...,n_m]$ in the tree. A match between hypothesis h_i and node n_i simply means that they have the same lexical name. The next step is to compare all sons of node n_m, $[n_{m+1,1},n_{m+1,2},...,n_{m+1,p}]$, with all hypotheses adjacent to hypothesis h_m, $[h_{m+1,1},h_{m+1,2},...,h_{m+1,q}]$[1]. If a match is found between node $n_{m+1,i}$ and some hypothesis $h_{m+1,j}$, then node $n_{m+1,i}$ (representing a unique path in the tree) is saved to be extended later. If the sequence of lexical items given by the path defines a lexical item at level l, an hypothesis is made at level i with begin- and end-segment numbers and rating obtained from the hypotheses $[h_1,h_2,...,h_m,h_{m+1,j}]$. Since different paths may define the same lexical item or different sequences of hypotheses may match the same path in the tree, duplicate hypotheses may be generated, differing only in rating. All duplicate hypotheses except the best-rated hypothesis are deleted.

5.2.3 Features of Recognition Unique to the Lower Levels

5.2.3.1 Segment Level to Sylpart Level

There are three trees between the segment level and the sylpart level: the segment-vowel tree, the segment-onset tree, and the segment-coda tree. Recognition begins with the segment-vowel tree, starting at the first segment of a syllable nucleus (segment 22 in Figure 5.1)[2]. The segment label hypotheses at this segment are matched with the first nodes of the tree. Since the segment patterns in the segment-vowel tree are not necessarily stored in a left-to-right sequence (see Section 4.3.3.2 on "nonsequential pattern storage"), the next nodes in the tree determine whether the segments to the right or left are compared next. Thus, the order in which the segments are compared is part of the pattern stored in the tree. A vowel hypothesized will have a beginning segment number equal to the leftmost segment compared and an ending segment number equal to the rightmost segment compared for its pattern.

Any segment immediately to the left of the first segment of a vowel hypothesis becomes a starting segment for searching the segment-onset tree. For this tree, segments are compared right-to-left (i.e., back in time) as the tree is searched. Segments for the segment-coda tree are compared in the forward direction, starting with any segment immediately to the right of the last segment of a vowel hypothesis.

---

1 To avoid p times q comparisons, the sons of each node and the hypotheses on each list of adjacent hypotheses are ordered by the lexical number of their lexical items. This permits at most p+q comparisons.

2 As described in Chapter 4, a syllable nuclei is a sequence of contiguous segments each of which has a segment class name beginning with a "V".

Unique to these segment-sylpart trees is the ability of one node in a tree to span more than one segment (see Section 4.3.2 on "segment pattern learning"). For example, a vowel with the segment pattern "[Y]" may be hypothesized to span segments 21 through 23 since a "Y" segment label is present in each segment with a good (i.e., numerically low) rating. This form of a dynamic programming algorithm permits Noah to use nonsegmented speech. Suppose the speech corresponding to segments 21 through 23 of the example had been divided into 10 milli-second samples and then labeled. If the "Y" label was rated well in each sample (as it was in each segment) the same vowel would be hypothesized to span the same part of the utterance. However, segmenting reduces the cost of recognition. In this case, only 3 segments are looked at rather than 140 (10 milli-second) samples (i.e., the time span between centi-seconds 125 and 139).

5.2.3.2 Sylpart Level to Syllable Level

The search of the sylpart-syllable tree begins at the vowels in each syllable region, continues with the onsets, and finishes with the codas. This is done in order to handle null onsets and null codas easily.

Vowel sequences are expanded during this search and treated as a separate sequence of sylparts. For example, the vowel sequence "IY R IY" is expanded into its parts. When the first "IY" begins the sylpart-syllable tree search, the "R" is used an optional coda -- the only other option being a null coda. When the second "IY" is used to start the sylpart-syllable tree search, the "R" is used as an optional onset -- the other option being a null onset. Any inital non-vowel in a vowel sequence (such as the "Y" of "Y UW") is appended to the end of adjacent onsets whenever such a joining results in a legal onset name. For example, the "Y" in the vowel sequence hypothesis "Y UW" spanning segments 21 through 23 is appended to the onsets ending at segment 20, producing onsets "M Y" and "N Y", both legal onsets. Similarly, any final non-vowel in a vowel sequence is appended to the begining of adjacent codas if it results in a legal coda name. Vowel sequences with more than one vowel produce syllables with special adjacency restrictions.

Consider the vowel sequence "IYAX". This vowel sequence produces the set of syllables characterized by the sequence: <some onset><the vowel "IY"> <a null coda> and the set of syllables characterized by <a null onset><the vowel "AX"> <some coda>. Syllables from the first set are adjacent on the right only to syllables in the second set. The syllable "M IY" in Figure 5.1 is adjacent on its right only to the syllable "AX N" since both are based on the vowel sequence "IYAX". (Their times overlap because each uses the complete segment span of the vowel sequence).

5.2.4 Parallel Recognition

Although a parallel recognition algorithm has not been implemented, the recognition algorithm given above permits a high degree of parallel processing. Matching the sons of a node in a tree with a set of hypotheses depends only on having matched the previous nodes in the tree and having the complete set of hypotheses. Thus, processes can work at the same time in different branches of the same tree in the same place in the utterance, or in the same tree in different parts of the utterance, or in different trees at different levels. One possible division of processing is to use one processor for each syllable region. In each region a processor would first hypothesize vowels, onsets, and codas, then hypothesize syllables, and finally search the syllable-word tree starting with all syllables in its region and continuing with other regions as the syllables became available. Depending on the number of processors, this scheme would decrease the recognition time for the typical utterance by more than an order of magnitude.

5.3 Rating of Hypotheses

There is no end to changing rating methods, adjusting thresholds, and generally trying to get optimal results from errorful input. The rating methods reported here have no proof of optimality but give the best results so far and seem to make sense. Ratings are used by the word hypothesizer to report the likelihood that a word was spoken at a particular place in the utterance and to limit the search of trees at each level. As has been stated, the rating of each hypothesis ranges from 0 to some upper limit, with 0 corresponding to a perfect score (similar to the negation of the log of likelihood probabilities). The rating of a hypothesis made up of a sequence of hypotheses at the next lower level is equal to the sum of the ratings of the lower hypotheses minus a normalizing value depending on the length of the sequence. This sum is computed as the path through the tree is searched. If at any node in the search this sum minus any possible future normalizing value becomes greater than a set threshold (different for each tree), the search of the tree at the node and beyond is pruned. Thus, searching for hypotheses that will be rated poorly is aborted early.

In the case of a sylpart, the final rating of the hypothesis also includes a context rating and a weight penalty. As described in Chapter 4, a set of context segment label pairs is stored in the tree for each segment pattern of each sylpart. These context segment label pairs are the top segment label hypotheses[3] found to the left and to the right of the segment pattern each time the pattern was learned for the sylpart. The

---

3 Subject to the modification described in Section 4.3.4 on "context learning".

context rating is computed during recognition by finding the stored pair of context segment labels which match best with the segments to the left and right of the place were the pattern is currently matched. The sum of the ratings of these best segment labels is divided by a factor based on the length of the segment pattern and then added to the rating of the sylpart hypothesis. The reasoning behind this computation is that the closer the current context for the segment pattern of a sylpart matches with some previously learned context, the more likely the segment pattern should be interpreted as representing the same sylpart. Also, the shorter the segment pattern, the more influence the context segments will have on it. In the case of a zero-length segment pattern, the rating of the context segment labels completely determines the rating of the hypothesis. An example of this is the onset "D" in the context of a nasal and a vowel as might occur in the word "Standing" -- (S T AE N - D IH NX).

Segment patterns:	[- PH]	[←]	[SH]	[- P]	...	Row Max
Onset sylpart: P	2	0	0	19	...	19
T	1	1	2	0	...	10
K	10	0	0	1	...	10
D Y	1	2	0	0	...	2
SH	0	0	16	0	...	16
...	...	...	...	...		
Column Sum:	16	23	34	22		

Figure 5.2: Sample Frequency Counts of Segment Patterns for Onsets

The weight penalty penalizes a sylpart hypothesis when a) it is based on a segment pattern which occurs much more frequently for other sylparts, and b) its empirically observed conditional probability is low for the pattern. Thus, the weight penalty attempts to distinguish between sylparts having the same segment pattern. Figure 5.2 shows a portion of a training frequency count matrix for the segment patterns of onsets. For instance, one can tell from this figure that the segment pattern "[- PH]" appeared twice for the onset "P" in the training. The numbers in the rightmost column give the number of times the most common segment pattern appeared for the onset of the row. The bottom row shows the total number of times each segment pattern appeared. Let R_i be the row maximum for the sylpart S_i, C_j be the column sum for the segment pattern P_j and F_{ij} be the frequency count of pattern P_j for sylpart S_i. The weight penalty, W_{ij}, for S_i and P_j is obtained by:

$$W_{ij} = \langle \textit{maximum weight penalty} \rangle * (1 - (F_{ij}/C_j \max F_{ij}/R_i))$$

The penalty W_{ij} is based on empirically observed $Prob(S_i|P_j) = F_{ij}/C_j$ limited by the ratio of F_{ij} to R_j. For example, Prob("D Y"|[←]) = 2/23, but since "D Y" occurs

infrequently in training, its most common occuring pattern occurs only twice. Thus, the pattern "[←]" receives no weight penalty for the onset "D Y". On the other hand, consider the pattern "[- PH]" for the onset "P". The Prob("P"|[- PH]) = 2/16 is small and the ratio of the occurrences of "[- PH]" for "P" to the maximum occurrences of any pattern for "P" is even smaller (= 2/19) so that the weight penalty is high. It is not very likely that pattern "[- PH]" will represent onset "P".

Rating method:	Segment pattern match	Including Context	Including Context and Weight Penalty
Vowel results:			
% correct	81%	81%	79%
Average rank	5.1	3.1	2.7
# hypotheses/syl.	22.7	19.5	13.3
Onset results:			
% correct	93%	93%	91%
Average rank	6.6	4.3	3.7
# hypotheses/syl.	19.0	16.6	13.1
Coda results:			
% correct	90%	90%	90%
Average rank	5.8	3.9	3.9
# hypotheses/syl.	16.8	15.0	12.8

Figure 5.3: Sylpart Recognition Results for 215 Syllables

Figure 5.3 shows the effect of including the context rating and the weight penalty with the rating of the segment pattern match for the recognition of sylparts for 215 syllables. The rank of a correct hypothesis is the number of incorrect competing hypotheses rated better than it, plus one. This rank value is averaged for all correct hypotheses for each type of sylpart to give the average rank. Both the addition of the context rating and the weight penalty gives the desired decrease in the average rank as well as eliminating many incorrect hypotheses. However, the weight penalty increased the rating for a few of the correct vowel hypotheses and onset hypotheses above the acceptance threshold so that they were eliminated. The addition of context decreases the average rank for the sylparts by about 35%. The addition of the weight penalty decreases the average rank another 10%.

5.4 Propagation of Segment Label Confusion During Recognition

A frequent lament of speech system designers is: "If only we had a better segmenter-labeler". Although such a desire is almost equivalent to a wish for cleaner and more easily recognized speech, it is clear that the word hypothesizer would certainly profit from a better segmenter-labeler. The effect of segment-label confusion on the word hypothesizer is shown here by using the measure of hypothesis confusion developed in Section 2.4. Note that this confusion measure says nothing about correct or incorrect hypotheses; it only measures the confusion (i.e., the uncertainty of information, in a nontechnical sense) of a set of hypotheses (segment-labels, sylparts, syllables, or words) based on the past performance of the segmenter-labeler. As is pointed out in Chapter 2, the confusion measure can be interpreted as measuring, for a set of competing hypotheses, the equivalent number of hypotheses rated the same as the best hypothesis. Figure 5.4 traces the confusion of the hypotheses from the segment level through 14 steps of the recognition algorithm up through the word level. The first number for each set of competing hypotheses gives the average confusion measured for the set at the particular point in the recognition algorithm. The number in parentheses gives the average number of hypotheses in the set. For example, at the bottom "14(98)" means that 98 segment labels are given as input to Noah for each segment (seven segments are represented by boxes) and on the average the ratings for the labels are such that the confusion is 14, i.e., the equivalent number of equally rated labels is 14. A brief description of the recognition step is given on the left of the figure and a symbolic representation is on the right. For example, on the average there are 13 segment-labels from the segmenter-labeler which pass a threshold (i.e., they have a good enough rating according to a threshold criterion). Their ratings are such that they have an average confusion of 6.

One task of the recognition algorithm is to reduce the confusion of the segment labels by applying the constraints of its description of words. We see from Figure 5.4 that the segment label confusion is reduced (or remains the same) in every step except for three. These three are: a) Step 2: joining the labels according to the sylpart syntax, b) Step 4: interpreting the segment patterns -- making sylpart hypotheses based on the patterns, and c) Step 8: joining the onset, vowel, and coda hypotheses to form syllables. In Step 2, the sylpart syntactical constraints do in fact reduce the confusion from what it would have been if every label of a segment could be joined with every label of adjacent segments to form patterns. In Section 2.4, we saw that the confusion of a set of hypotheses at one level formed from all combinations of the hypotheses in two sets of adjacent hypotheses at a lower level equalled the product of the confusions for the two sets. Generalizing this to more than two sets, we can estimate the confusion of set of hypotheses at one level based on the confusion of the sets at a lower level when no

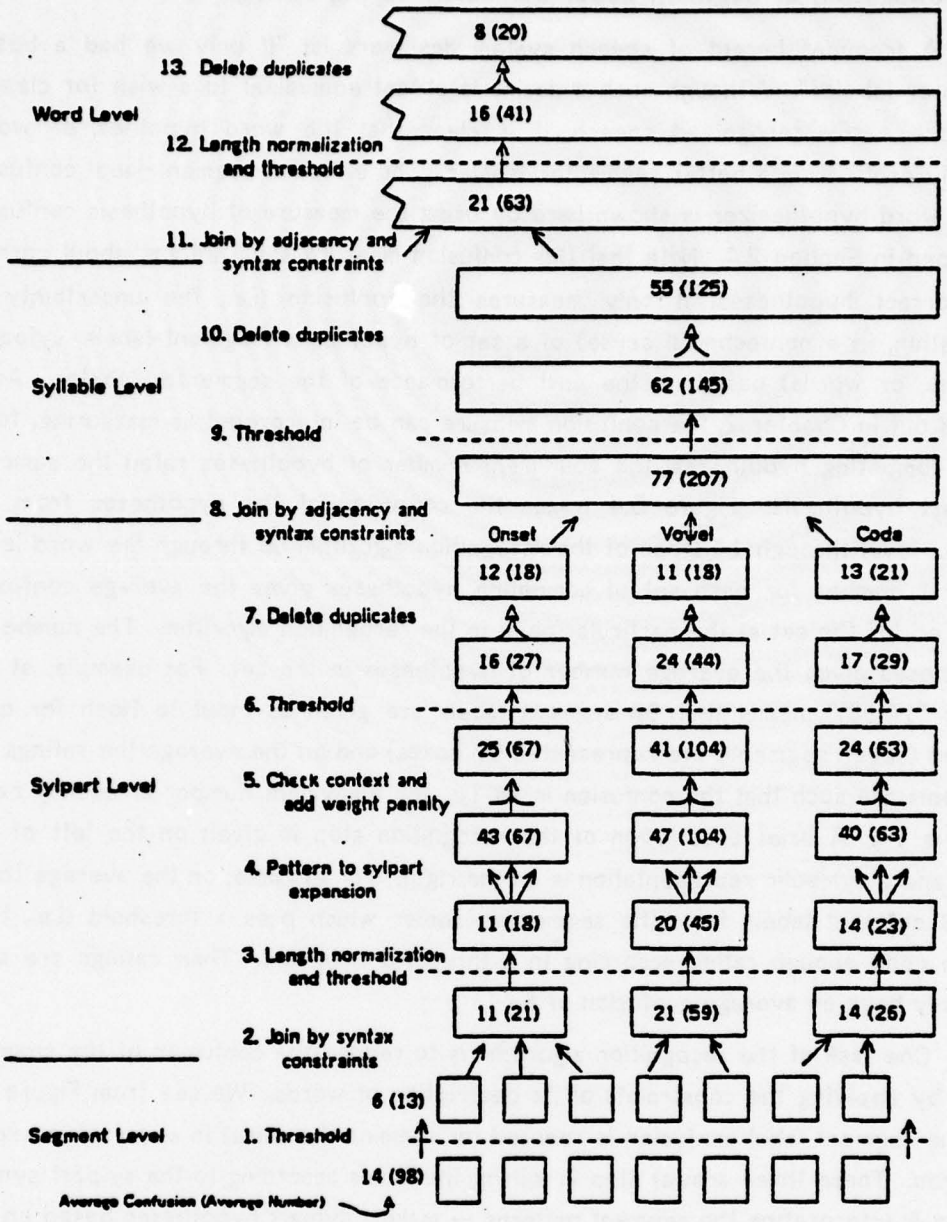


Figure 5.4: Propagation of Confusion During Recognition

syntactic constraints are applied. For example, the average number of segments for an onset sylpart is 1.7. The expected average confusion for a set of competing onsets when no syntactical constraints are applied is the measured confusion of a segment (6) raised to the 1.7 power, or 21 (= $6^{1.7}$). Comparing this value to the actual average confusion of 11 measured for an onset, we see that the learned syntactic constraints

reduced the confusion by about one-half. In the same way, the expected confusion for Step 8 (for the syllables) is approximately 400 (= $12^{2.4}$). The adjacency and syntactic constraints, used to put the onset, vowel, coda hypotheses together, reduce this number to 77.

The increase in the average confusion for Step 4 of the recognition algorithm is easily explained. Since one segment pattern may represent several sylparts, an increase in the number of competing hypotheses occurs when the algorithm makes sylpart hypotheses from the recognized segment patterns. For example, the onsets "P" and "T" share the segment pattern "[- PH]". When that pattern is recognized, the onset "P" and the onset "T" are each hypothesized. The increase in competing hypotheses causes a corresponding increase in the average confusion. The increased confusion introduced by the multiple interpretations of segment patterns in this step is reduced in part by the use of context and the weight penalty in the next step.

There are two sources of confusion for the algorithm. The first is from the segment-label input to the algorithm; the second is from the multiple interpretations of the segment patterns, as seen in Step 4. The second source is due to the ambiguity of speech stored in the segment-sylpart trees and acquired from the segmented and labeled training utterances.

The final result of the recognition algorithm according to the confusion measure is 20 competing word hypotheses per utterance word with ratings such that the equivalent number of equally likely word hypotheses is 8.

We have used the confusion measure to give another view of the recognition algorithm. Much more could be done with this measure. For example, it could be used to see the effect of adjusting various thresholds throughout the algorithm; it could be used to see where in the algorithm larger vocabularies cause greater confusion; and it could be used to analyze the effect of training. Time did not permit a more thorough application of the measure.

Chapter 6: Results and Analysis

6.1 Introduction

This chapter is separated into three main sections: performance and runtime characteristics, analysis of performance, and a comparison with two other word hypothesizers. Before performance is discussed we need to describe how performance is measured and the conditions under which Noah was trained and tested.

The difficulty with reporting performance for a word hypothesizer is that the measure of performance is closely connected with the characteristics of the speech system in which the hypothesizer is used. For example, in a speech system which uses only the best correct bottom-up word hypothesis to begin a top-down (grammar restricted) search for the rest of the words of the utterance, the relevant performance is the rating of the best correct word hypothesis relative to the ratings of all other hypotheses.[1] The goal, in this case, is always to rate a correct word hypothesis better than all other hypotheses and to minimize the number of other hypotheses. Performance has been measured for Noah with a more bottom-up speech system in mind. The goal for a word hypothesizer in such a system is to hypothesize all of the correct words and no others. (If such a goal were reached, the rest of the system would have little to do.) Thus, the relevant performance is given by 1) the number of words hypothesized, 2) the number of correct words hypothesized, and 3) the ratings of the correct hypotheses relative to the ratings of the incorrect ones. The next section describes in detail what measurements of performance are used.

6.1.1 Measurements of Performance

6.1.1.1 Word Accuracy and Average Rank

The first question to answer is "What is a correct word hypothesis?" All test utterances (105) were hand segmented into words -- that is, the words of the spoken

1 The performance of Noah will be measured by this method when it is compared to the Lexical retrieval component of the HWIM system - a system using the above strategy for recognition.

utterance were given begin and end segment numbers by inspecting the segment labels and the speech waveform. (Of course, this hand-segmentation is used only for performance analysis -- Noah cannot access it while doing its hypothesization.) A word hypothesis matching a hand-defined correct word in name and having a begin segment within 2 segments of the "correct" begin segment and an end segment within 2 segments of the "correct" end segment is considered to be a correct word hypothesis. The ratings of a correct word hypothesis relative to the incorrect ones is measured by computing the rank of the correct hypothesis. The rank of an hypothesis is defined to be the number of competing hypotheses rated better than it, plus one-half the number of competing hypotheses rated the same as it, plus one (i.e., a rank of one is the best rank - no hypotheses rated better). Two hypotheses are considered to compete if the amount of their overlap in time is greater than one-half the duration of the shorter hypothesis. For example, in Figure 6.1 hypothesis A competes with hypotheses C, D, and E (and vice-versa) but not with B. This is definition is somewhat arbitrary and it may result in calling two hypotheses "competing hypotheses" which a speech system would not consider as competing. However, it is the definition used by POMOW and is used here to permit a comparison of the word hypothesizers.

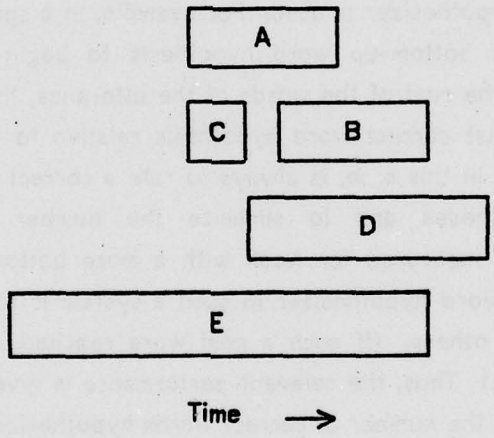


Figure 6.1: Competing and Noncompeting hypotheses

Added to the above tests for a correct hypothesis is the restriction that its rank be less than or equal to 20. Though this threshold of 20 is somewhat arbitrary, it was thought that no speech system could afford to look deeper than about 20 words for the correct word in the bottom-up word hypotheses at any place in the utterance. This restriction is similar to stating that no more than 20 words are hypothesized for each utterance word. The performance of Noah can now be given by two numbers: the

percent of the utterance words correctly hypothesized and the average rank of the correct word hypotheses. The goal is therefore 100% word accuracy at an average rank of 1.

6.1.1.2 Average Efficiency

In an attempt to reduce the measure of performance to one number which could be used to monitor the progress in developing the word hypothesizer and to compare it across different vocabulary sizes, the measure of "average efficiency" was defined. The average efficiency measures a weighted word accuracy of the hypothesizer by weighting each correct word hypothesis according to its rank. The average efficiency for a set of utterances is given by:

$$\text{Average Efficiency} = 1/n * \sum_{1 \le i \le n} 1/R_i,$$

expressed as a percent, where R_i is the rank of the ith correct word hypothesis and n is the total number of words in the test utterances. If the ith correct word is missed, its rank is taken to be infinity. This equation is equivalent to computing a weighted accuracy by counting all the correct hypotheses at rank 1, 1/2 of the correct hypotheses at rank 2, ... and 1/n of the correct hypotheses at rank n. Thus, the average efficiency varies from 0% for no correct hypotheses to 100% for a perfect word hypothesizer. The term "efficiency" was chosen because the measure is the ratio of work output (one correct hypothesis) to the work input (producing another hypothesis) at each place in the utterance if the word hypothesizer is used as a word generator. A word generator produces the next best-rated word hypothesis at any place in the utterance as requested by the rest of the speech system. Averaging this ratio over a set of utterances gives the "average efficiency".

Two criticisms can immediately be made about this measure. First, it says nothing about the number of hypotheses made. This can be answered by remembering that at no time do more than the top 20 ranks need be considered in searching for the correct word. Second, the average efficiency is a harsh performance measure. It puts a lot of importance on hypothesizing the correct word in the top rank, but whether 2 correct word hypotheses at rank 2, or 5 correct word hypotheses at rank 5 are better than only 1 correct word hypothesis at rank 1 depends on the speech system using the word hypothesizer and on the complexity of the language. At any case, the average efficiency measure is insensitive to correct hypotheses in the higher ranks. This is not true for the word accuracy or average rank measures. For example, if 70 out of 100 words are hypothesized correctly at an average rank of 3 and an average efficiency of 40% and then one more correct hypothesis is made at rank 20, the word accuracy increase by 1% and the average rank by .24, but the average efficiency increases only by 0.05%.

For completeness, we will also show the accumulative word accuracy at the best 5 ranks. That is, the word accuracy when only the first rank is considered, when only the first two ranks are considered, ... and finally, when only the first five ranks are considered.

6.1.1.3 Summary of Performance Measures

We summarize here some of the terms and performance measures given above; the performance measures are underlined:

Correct Word Hypothesis: A word hypothesis which matches an utterance word in name and in position in the utterance. The position matches if the begin and end segments of the hypothesis are each within two segments of the corresponding begin and end segments of the utterance word.

Competing Hypotheses: Two hypotheses whose overlap in time is greater than one-half the duration of the shorter hypothesis.

The Rank of an Hypothesis: The number of competing hypotheses rated better than the hypothesis plus one-half the number rated the same plus 1.

Word Accuracy: The percent of utterance words for the test data having Correct Word Hypotheses at a rank less than or equal to 20.

Average Rank: The average rank of the Correct Word Hypotheses for the test data.

Average Efficiency: A weighted Word Accuracy computed by summing 1/(The Rank of a Correct Word Hypothesis) for all Correct Word Hypotheses and dividing by the total number of utterance words.

Word Accuracy for the best M ranks: The Word Accuracy if hypotheses are limited to the best M ranks. (Word Accuracy above is for the best 20 ranks.)

6.1.2 Training and Testing Conditions

Noah was trained on 174 utterances (about 1600 syllables) that had been hand-segmented into sylparts, as described in Chapter 4. (This set was repeatedly divided to obtain the smaller training sets). A different set of 105 utterances (705 words) made up the test sentences. The test sentences had been looked at only to hand-segment them into words before testing. All sentences were spoken by the same speaker in a quiet room with a close-speaking microphone in sessions spread out over a period of several months. The word hypothesizer was adjusted for best performance using the

1000-word vocabulary[2] and then tested on the other size vocabularies without further adjustment. A 500-word vocabulary was made from a subset of the 1000-word vocabulary which still included the 268 words of the test sentences. The larger vocabularies were formed by adding to the 1000-word vocabulary subsets (every Nth word) of the 20,000-word dictionary. No word in this dictionary was included in the test vocabularies if it had already appeared in the 1000-word vocabulary or if it included sylparts which did not appear in the training utterances. Thus, in all tests, each word in the vocabulary was unique and each had the potential of being hypothesized.

6.2 Performance and Runtime Characteristics

This section contains a description of Noah's performance; an analysis of that performance is given in the following section.

6.2.1 Performance versus Word Vocabulary Size

The graphs of Figure 6.2 give Noah's performance by three measurements (word accuracy, average correct word rank, and average efficiency) as a function of vocabulary size. Word accuracy is seen to drop from 73% to 58% as the vocabulary increases from 500 words to 19,000 words. At the same time the average rank of the correct word hypotheses climbs from 2.6 to 5.8 for the same increase in vocabulary size. The efficiency measure shows a smooth decline in performance which is approximately logarithmic in vocabulary size. (A plot of a logarithmic curve is also given).

Figure 6.3 gives the word accuracy in the top 5 ranks. From these graphs it is easy to see the effect on the accuracy of limiting Noah to hypothesizing only the best M words at each point in the utterance (for M=1,2,...,5).

6.2.2 Performance versus Training Sample Size

The performance of Noah, again given by three measures, is plotted against the total number of sylpart paths learned from different sizes of training sets in Figure 6.4 for the 1000 and the 8000-word vocabularies. The number of sylpart segment paths rather than the total number of sylpart samples is displayed on the x-axis to show the effect that new information has on the performance. For both vocabularies the word accuracy as well as the average rank of the correct hypotheses is seen to rise as the

---

2 The "1000-word" vocabulary is the one used by the Hearsay II system. Actually this vocabulary is made up of 1011 words. The other vocabularies consisted of 508, 2013, 4020, 8032, 16,025, and 19,008 words.

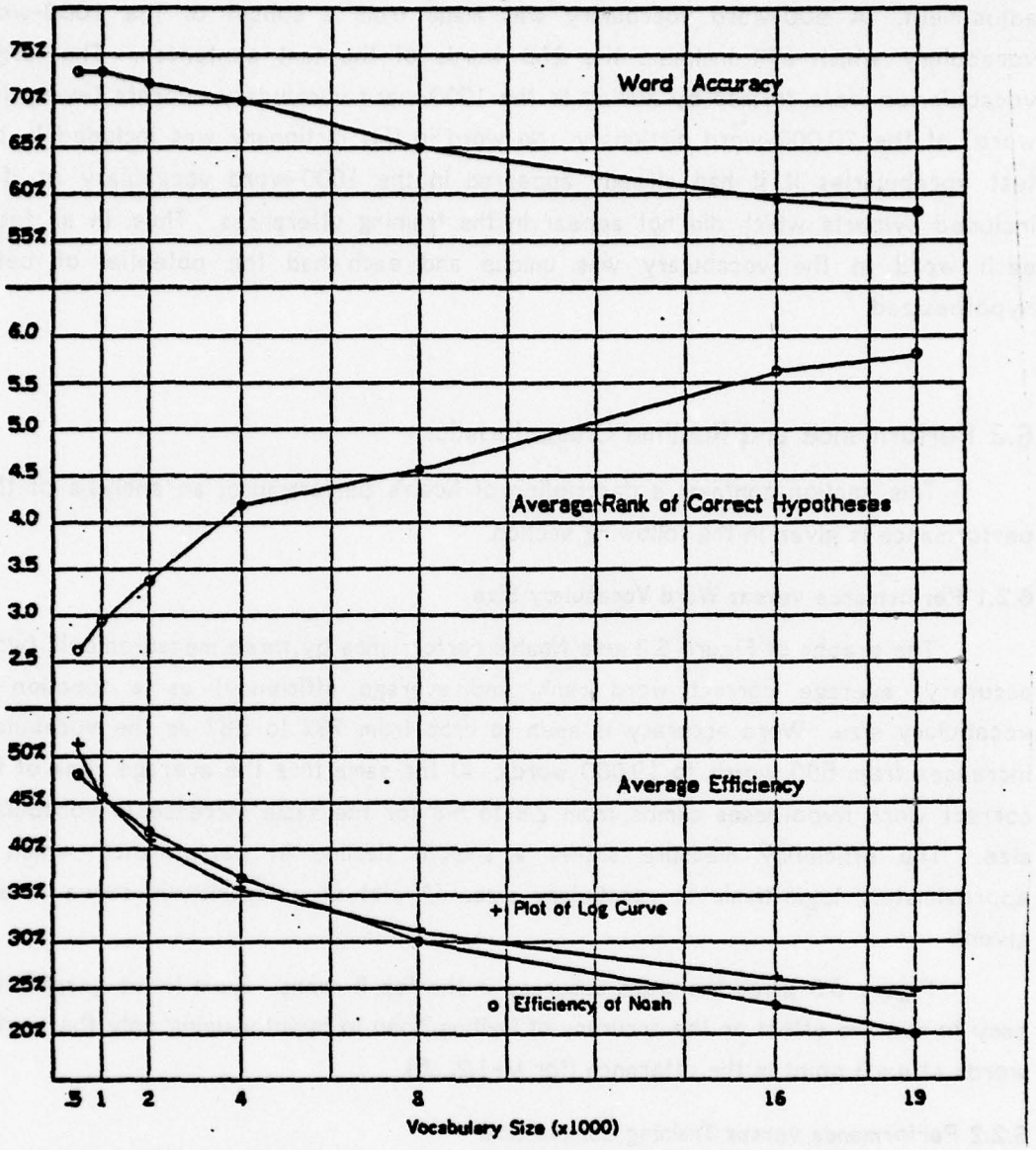


Figure 6.2: Performance of Noah versus Vocabulary Size

training increases. This results in a gradual increase in the average efficiency of Noah for the 1000-word vocabulary but no increase in the average efficiency for the 8000-word vocabulary after about 1080 sylpart segment paths have been learned (explained in Section 6.3.2).

6.2.3 Computation Costs versus Vocabulary Size

The Computation Cost given as millions of instructions executed per second of speech (MIPSS) is plotted in Figure 6.5 as a function of the log of vocabulary size for

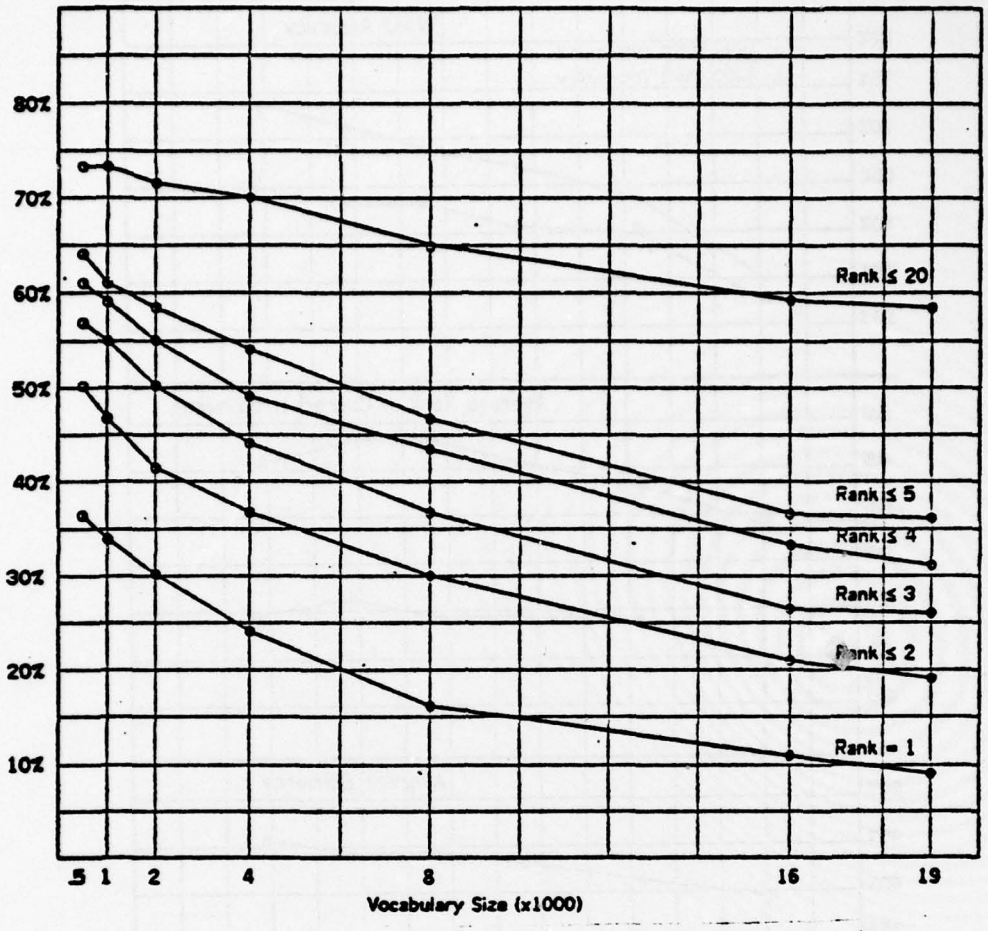


Figure 6.3: Accuracy in Top M Ranks versus Vocabulary Size

different subparts of the word hypothesizer. The total cost increases at a logarithmic rate from 2.4 MIPSS to 6.6 MIPSS as the vocabulary increases from 500 words to 19,000 words. (This is an increase of about 0.75 MIPSS per doubling of the vocabulary.) The time for processing speech times real-time (i.e., the computation time divided by the time to speak the utterance) is shown by a scale to the right for the machine used in these tests -- a Digital Equipment Corporation PDP-KL10 which is about a 1.3 million instruction per second machine. The effect of training on total computation costs is small. Over the range of training sets used, the change in computation cost was about 0.5 MIPSS.

6.2.4 Breakdown of Storage Costs

The Noah word hypothesizer (not including segment pattern learning program or the dictionary processing program) requires 51K of 36-bit words of storage containing code for recognition, debugging, analysis and statistics, and runtime constants and

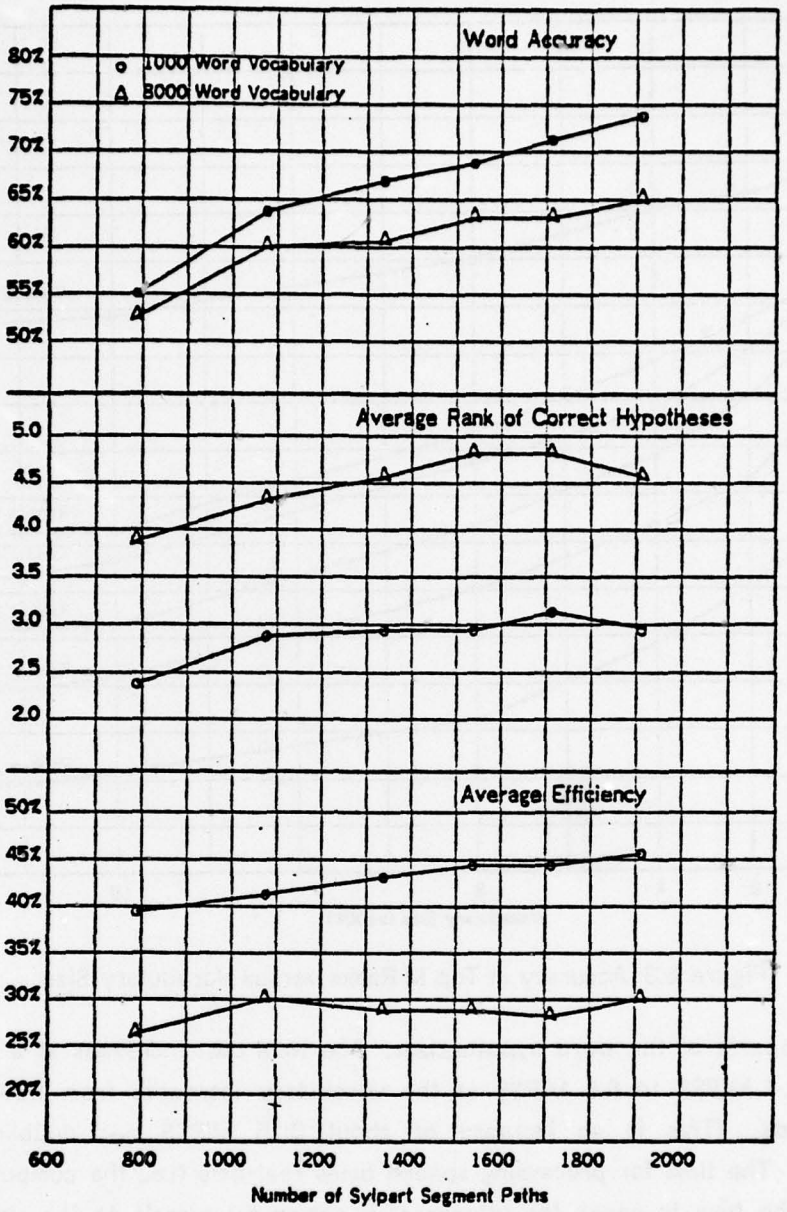


Figure 6.4: Performance of Noah versus Training

variables. A variable amount of storage is used for the tree structures containing the knowledge of the system and for the structures containing the hypotheses at segment, sylpart, syllable, and word levels. The size of this storage depends on the amount of training, the size of the word vocabulary, and the length of the utterance. The breakdown of storage costs for the hypothesizer working on a 51 segment utterance (about 3 seconds) with a 1000-word vocabulary and the full training set used here (174 training utterances) is as follows:

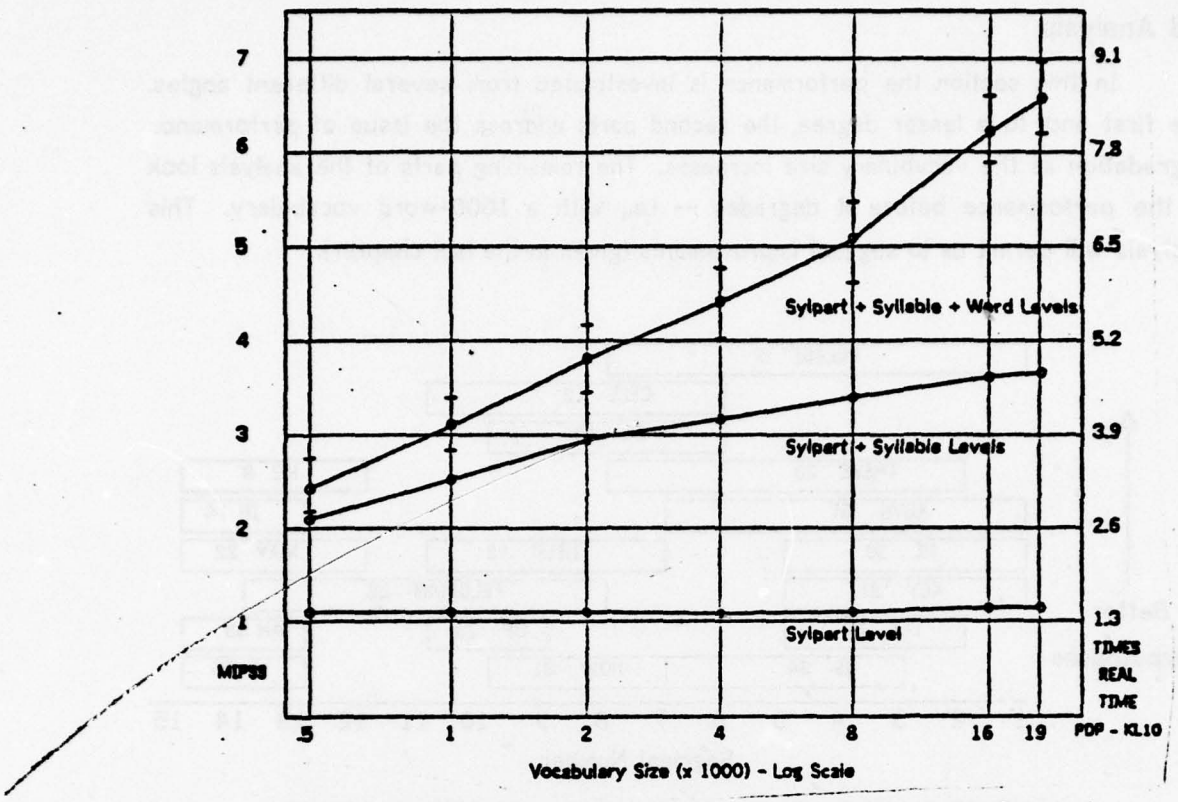


Figure 6.5: Computation Cost versus Vocabulary Size

	1000-word vocabulary	19,000-word vocabulary
Code and fixed storage:	51K	51K
Hypotheses for utterance:	13K	13K
Segment-sylpart tree (training dependent):	12K	12K
Sylpart-syllable and syllable-word tree (vocabulary dependent):	11K	148K
Total storage requirement:	87K	224K

Note that the storage cost for the vocabulary-dependent knowledge increases to 148K (13 times greater) for the 19,000-word vocabulary. It would be quite easy to implement a paging algorithm for the syllable-word tree, the major part of the 148K. (The tree was implemented with a paging algorithm in mind, but paging was not found to be necessary with the amount of primary memory available.) Storage costs as a function of training and vocabulary size have been given in Chapter 3.

6.3 Analysis

In this section the performance is investigated from several different angles. The first and, to a lesser degree, the second parts address the issue of performance degradation as the vocabulary size increases. The remaining parts of the analysis look at the performance before it degrades -- i.e., with a 1000-word vocabulary. This analysis will permit us to suggest improvements (given in the last chapter).

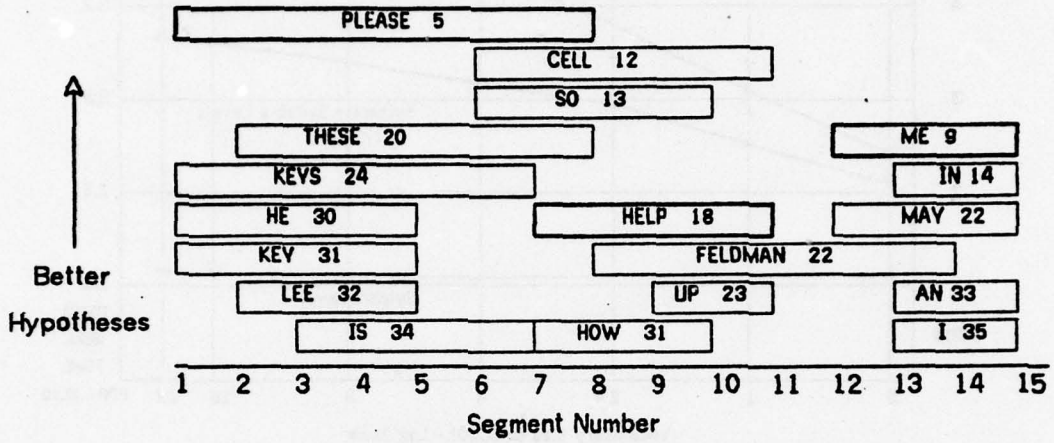


Figure 6.6A: All Competing Hypotheses for "Please Help Me", 1000-Word Vocabulary

6.3.1 Effect of Vocabulary Size on Performance

Performance degrades, as expected, as more words compete for hypothesization in the utterance. However, by one measure -- the average efficiency -- the performance degrades only by a factor proportional to the log of the vocabulary size over the range of vocabularies tested. Figures 6.6A and 6.6B show how the performance of an almost perfectly recognized utterance using the 1000-word vocabulary drops when the 16,000-word vocabulary is used. The average rank increases from 1.7 to 4.2, while the average efficiency drops from 78% to 70%.

In most cases, for this example, the incorrect competing words from the 16,000-word vocabulary are probably those which a human would recognize if he heard only the corresponding part of the acoustics. What is missing from the word hypothesizer is the ability to recognize syllable and/or word boundaries. (The possibility of doing this is discussed in Chapter 7).

Why is the performance degradation as measured by the average efficiency approximately proportional to the logarithm of the vocabulary size? Though we can not

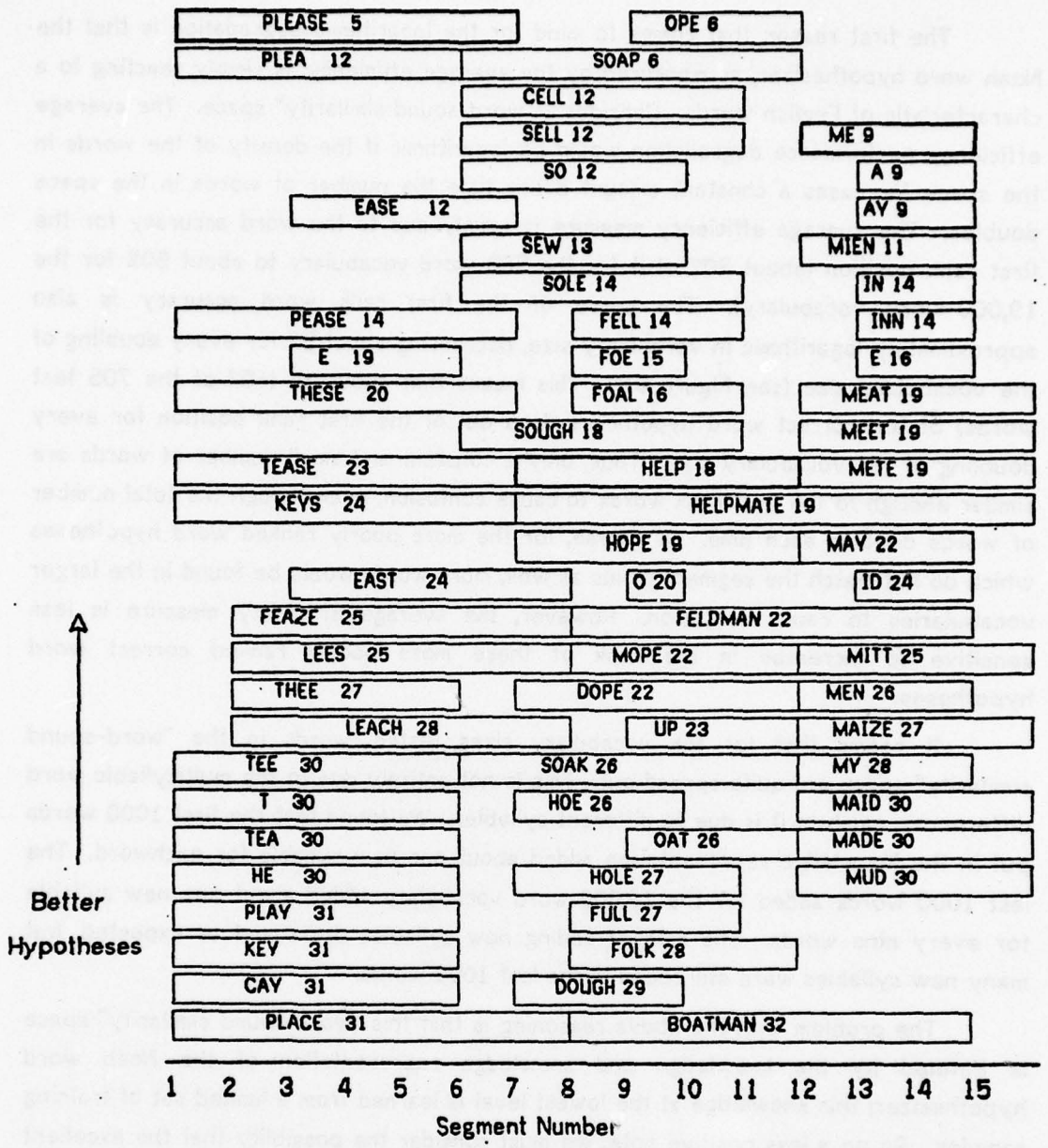


Figure 6.6B: The Top Competing Hypotheses for "Please Help Me," 16,000-Word Vocabulary

answer this question completely, we can suggest some reasons. However, we must caution that the average efficiency is only one measure of performance; some other measure might give a different rate of performance degradation. In particular, it is not known how a speech system's performance might react to a logarithmic degradation in the average efficiency of a word hypothesizer for the range of vocabularies tested.

The first reason that comes to mind for the logarithmic degradation is that the Noah word hypothesizer, as measured by the average efficiency, is simply reacting to a characteristic of English words. Consider a "word-sound similarity" space. The average efficiency performance degradation would be logarithmic if the density of the words in the space increases a constant amount every time the number of words in the space doubles. The average efficiency measure is mainly due to the word accuracy for the first rank position (about 80% of it for the 500-word vocabulary to about 50% for the 19,000-word vocabulary). The curve of the first rank word accuracy is also approximately logarithmic in vocabulary size, decreasing about 5% for every doubling of the vocabulary size (see Figure 6.3). This means that about 35 (=5% of the 705 test words) of the correct word hypotheses drop out of the first rank position for every doubling of the vocabulary size. Thus, only a constant and small number of words are similar enough to the first rank words to cause confusion, even though the total number of words doubles each time. Of course, for the more poorly ranked word hypotheses which do not match the segment labels as well, more words would be found in the larger vocabularies to cause confusion. However, the average efficiency measure is less sensitive to increases in the rank of these more poorly ranked correct word hypotheses.

It seems that for the vocabulary sizes tested, words in the "word-sound similarity" space are quite spread out. This is not entirely due to the multisyllabic word differences; much of it is due to different syllables. We found that the first 1000 words put in the knowledge representation added about one new syllable for eachword. The last 1000 words added for the 19,000 word vocabulary added about one new syllable for every nine words. The rate of adding new syllables decreased as expected, but many new syllables were still found in the last 1000 words.

The problem with the above reasoning is that this "word-sound similarity" space is defined by the knowledge and knowledge representation of the Noah word hypothesizer; this knowledge at the lowest level is learned from a limited set of training samples. So on a less positive note, we must consider the possiblity that the excellent logarithmic degradation as the vocabulariy size is increased is because the training and test utterances were both based on the 1000-word vocabulary. Although every sylpart in the 19,000-word vocabulary had training samples, one might suspect that the the number of training samples was biased towards those sylparts appearing in the 1000-word vocabulary, or in particular, toward those sylparts appearing in the 268 words of the test utterances. This is partly true. However, the bias in number of training samples is not very great: using the number of training samples for the least-trained sylpart of each word as a measure of the amount of training for the sylparts of a word, the average number of sylpart training samples for the words in the 268-word

vocabulary (the words of the test utterances), the 1000-word vocabulary, and the 16,000-word vocabulary was 25, 19, and 17, respectively. We suspect that a more prevalent bias is due to the context in which the sylparts were learned. Section 6.3.5 discusses how word accuracy is effected by training on the words of the test sentences as opposed to training on the sylparts of the words. Including a test word in the training guarantees that its sylparts will be learned in the appropriate context.

At any rate, we believe that any bias in the training affects only the rate of the logarithmic performance degradation, not the fact that it is logarithmic. The average efficiency curve is seen to be approximately logarithmic over the range of vocabulary sizes from 2000 words to 19,000 words. These words were randomly chosen from one 20,000-word dictionary so that each vocabulary has the same (possible) training bias.

We attribute the logarithmic increase in the computation cost with the vocabulary size to the tree searching done by the recognition algorithm. The cost of searching in trees storing information like that found in the sylpart-syllable and syllable-word trees is logarithmic in the number of terminal nodes. (See [Knuth - 1973], Vol. 3, pp. 499.) We believe the rate of this logarithmic increase has been reduced by using two levels of trees to store the dictionary knowledge. (The segment-sylpart tree is independent of the vocabulary size, as is indicated by the constant computation cost shown for this tree in Figure 6.5.)

6.3.2 Effect of Training on Performance

Figure 6.4 has shown that more training of the type used here is not the answer to better performance for larger vocabularies. Let us consider each sylpart segment pattern to be a point in some D-dimensional space with some distance metric. We can view the hypothesizer as if it used the nearest-neighbor algorithm; i.e., its job is to find the closest labeled point (corresponding to a learned segment pattern) to a new unknown point (corresponding to a segment pattern in the test sentences) in order to identify the new point. By training on more utterances, new labeled points are added to the space for labels which have little or no training, but at the same time many points are added to the space for frequently occuring labels. As bad samples occur, the volume covered by the points of these labels increases, causing more and more confusion during recognition. The possibility for more confusion increases with vocabulary size. This explains why the performance of Noah, according to the average efficiency measure, reaches a maximum with less training for the larger 8000-word vocabulary.

Two features of Noah are designed to counteract the above problem: pattern merging (discussed in Section 4.3.2) and the weight penalty (discussed in Section 5.3). It is possible that these heuristics need to be adjusted for larger vocabularies. A third

solution for this problem is selective training -- that is, learning additional segment patterns only for the sylparts for which errors occur. This can be done either by recording, hand-segmenting, and training on phrases containing the sylparts needing training or by automatic learning. We will discuss the potential for automatic learning in Chapter 7.

The anomaly in the average rank plots -- the drop for the final data points -- can be explained perhaps by something like selective training. For the 8000-word vocabulary, the learning of 200 more sylpart segment paths (from about 1710 to 1910 segment paths) resulted in a an improvement of 0.3 in the average rank. About 20% of this drop (0.06) is due to 21 new correct word hypotheses at an average rank of 6.4 and a loss of 7 correct word hypotheses at an average rank of 14. The remaining 80% (0.24) is due to an overall shift in the rank of the word hypotheses which were correct for both training sets. The 20 utterances[3] containing these 200 new segment paths contain a higher percentage of the words used in the testing utterances than the other training utterances.

6.3.3 Error Analysis for Sylpart Recognition

In a study of 215 syllables (from 20 utterances), accuracies for vowel, onset, and coda recognition were found to be 79%, 91%, and 90% respectively (see Figure 5.3). In 155 of these syllables (72%), all sylparts were recognized. In the remaining 60 syllables, errors were due to:

Errors	Number	Part of Total
No syllable nucleus found:	5	8%
Vowel-sequence not recognized:	17	28%
Vowel (and perhaps onset or coda) not recognized:	25	42%
Vowel recognized but onset and/or coda not recognized:	13	22%

Missing a sylpart always results in a missed word. Thus, the most likely cause (70% of the time) of not recognizing a word is that one of its vowels (appearing alone or as part of a vowel sequence) was not recognized. We expect that the errors due to vowel recognition could be cut in half by more careful training and using more information in the recognition algorithm (such as syllable stress), as suggested in Section 7.4.1. However, the variability found in the vowel pronunciations due to prosodics, context, carelessness of the speaker, etc., will always make vowels difficult to recognize bottom-up.

---

3 Set LLA; see Appendix D for a list of the training and test utterances.

Syllable nuclei were usually missed for a syllabic /N/'s as found in the final syllable of "written" and the final syllable of "hasn't". It would be quite easy to add to the segmenter-labeler tests for correctly recognizing these kinds of syllable nuclei.

6.3.4 Effect of Word Length on Word Accuracy

For the 1000-word vocabulary, the following was observed for words with different numbers of syllables:

Word length in syllables:	1	2	3	≥4
# words (% of total):	366 (52%)	216 (30%)	85 (12%)	38 (5%)
# recognized (% of column):	301 (82%)	155 (71%)	48 (56%)	13 (34%)
# words missed (% of tot.):	65 (9%)	61 (8%)	37 (5%)	15 (2%)
# hypothesized (% of total):	12111 (88%)	1430 (10%)	160 (1%)	22 (0%)
Prob. hypothesized word is correct for length:	.02	.11	.30	.59

Although the longer words contribute a smaller part of the total error, they are more apt to be correct (e.g., 13 out of 22 (59%) of the four or more syllable word hypotheses are correct). Also, long words are more often strong content words; that is, they carry the most important information of the sentence. The short monosyllabic words include "a", "the", "an", "of", etc., which do not carry much of the meaning of a sentence. Thus, it is clear that more emphasis on hypothesizing longer words would pay off.

6.3.5 Word Training versus Sylpart Training

One of the features of Noah which makes large vocabularies possible and at the same time permits changing the segmenter-labeler with a minimum of effort is the learning of segment patterns at the sylpart level rather than some higher level (such as the word level). The claim was that by properly handling coarticulation effects at the sylpart level, Noah could recognize any word in its vocabulary, even if the word had not appeared in its training. When it was noticed that the word accuracy for words not occuring in the training sentences was only 50% compared with an overall accuracy of 73% (using the 1000-word vocabulary), this claim came into question. However, the amount of sylpart training for these words must be studied before judgment is passed.

Figure 6.7 shows the accuracy for the words of the test sentences grouped in columns by the number of occurrences in the training sentences and grouped in rows by number of sylpart training samples for the sylpart of each word having the minimum amount of training. For each word, the sylpart with the minimum number of training samples is thought to be the most likely sylpart missed when attempting to recognize that word. The range for this number is on the left-hand side of the table. The groups

Number of Occurrences of Word in Training						Σ
		0	1→3	4→10	11→19	≥20	≥0
Number of Occurrences of Least-Trained Sylpart of Word in Training	1→7	21/55 38%	16/38 42%	5/10 50%	0/0 -	0/0 -	42/103 41%
	8→14	19/40 48%	24/43 56%	27/34 79%	7/8 88%	0/0 -	77/125 62%
	15→21	4/13 31%	12/24 50%	9/12 75%	30/35 86%	17/19 89%	72/103 70%
	22→35	23/34 68%	24/29 83%	24/28 86%	12/15 80%	0/0 -	83/106 78%
	≥36	12/15 80%	19/22 86%	60/66 91%	80/86 93%	72/79 91%	243/268 91%
	Σ ≥1	79/157 50%	95/156 61%	125/150 83%	129/144 90%	89/98 91%	517/705 73%

Figure 6.7: Word Accuracy as a Function ofWord Training and Sylpart Training

were chosen to give about the same number of samples along each dimension. This table seems to indicate that both factors, the amount of sylpart training, and the frequency of word training, contribute independently to word accuracy. Testing on the same words used in training guarantees that the sylparts are learned in the appropriate context. Thus, training on the same word generally gives better results than just training on the sylparts of the word. However, word accuracy is high for those words not appearing in training but having a large number of training samples for their sylparts.

6.3.6 What Words Should be Hypothesized?

It is generally thought that certain words should not be hypothesized by a bottom-up word hypothesizer. Usually these words are the "small function" words of a sentence such as "the", "and", "a", and "of", which do not give much clue to the content of the sentence and are often hypothesized incorrectly since they often appear as

subsets of other words. In testing Noah, all words were permitted to be hypothesized. At this point we can look at how the performance changes if certain words are eliminated from the hypothesizers vocabulary. This will be done without regard to the "content-value" of a word (i.e., how much the correct hypothesization of the word would constrain the rest of the words in the sentence).

Obviously, the current performance could be improved trivially by a post hoc elimination of all words that were not hypothesized correctly anywhere in the test utterances. Instead, considering only words which were hypothesized correctly at least once (177 words out of a possible 268), Table 6.1 shows the 15 worst words when the words are ordered by the number of times a word hypothesized correctly divided by the number of times it is hypothesized.

Word	# in Test	# times Correct	# times Hypothesized	Avg. Rank of Correct	# times better than Correct
ED	1	1	258	10.8	12
UP	2	1	238	4.8	23
AN	4	4	616	2.5	65
ART	1	1	127	12.8	9
IT	3	3	422	1.7	23
A	5	3	328	2.7	15
I'D	1	1	92	1.8	7
OR	1	1	188	3.8	4
DATE	1	1	88	5.8	4
AND	3	3	235	5.7	17
DID	3	2	148	4.8	3
KEN	1	1	67	1.8	3
LEE	1	1	61	5.8	3
ONE	1	1	62	3.8	1
BAY	1	1	58	8.8	1

Table 6.1: Worst 15 Words ordered by
(# times Correct) / (# times hypothesized)

Similarly, Table 6.2 shows the 15 worst words when ordered by the number of times the word was incorrectly hypothesized better than the competing correct hypothesis. If the top 6 words of Table 6.1 are eliminated, 14% of the total hypotheses (13723) of the 105 tests utterances would be eliminated but only 1.8% of the correct hypotheses. If the top 5 words of Table 6.2 are eliminated, the average rank drops from 2.9 to 2.4. However, these words occur frequently in the test utterances so that eliminating them reduces the word accuracy from 73% to 65%.

Clearly, it is the words with very little acoustic constraint which cause most of the incorrect hypotheses. We found that 1% of the 1000-word vocabulary (the ten words: "an", "at", "in", "it", "the", "of", "a", "to", "done", and "Ann") accounts for almost 30% of the incorrect hypotheses. Vocabulary words can be divided into four (fuzzy) groups as follows:

Word	# in Test	# times Correct	# times Hypothesized	Avg. Rank of Correct	# times better than Correct
AN	4	4	616	2.5	65
IN	9	9	436	2.6	37
OF	14	14	399	3.5	34
ANY	23	21	102	1.3	26
IS	9	8	229	3.1	24
UP	2	1	230	4.0	23
IT	3	3	422	1.7	23
THE	18	15	485	3.3	19
AND	3	3	235	5.7	17
DOES	5	4	124	2.5	16
A	5	3	328	2.7	15
ARE	19	17	189	4.5	13
TO	13	12	315	3.5	12
I	11	11	245	3.5	12
ED	1	1	258	10.8	12

Figure Table 6.2: Worst 15 Words Ordered by Number of Times Rated Better than the Correct Word

		Acoustic Constraints	
		Weak	Strong
Syntactic/Semantic "Content-Value"	Weak	I. ("An")	II. ("Also")
	Strong	III. ("Ed")	IV. ("Abstracts")

An example of a member of each group is given in parentheses. A word hypothesizer has no trouble with hypothesizing words from Groups II and IV; the strong acoustic constraints reduce the number of misses for these words. The problem is with the words of Groups I and III. As shown in the figures above, these words are often hypothesized incorrectly. We suggest that the word hypothesizer use variable acceptance thresholds for words in these groups so that the words in Group I are hypothesized only if they have "very good" ratings and words of Group III are hypothesized only if they have "fairly good" ratings. Of course, "very good" and "fairly good" would have to be defined. The acceptance threshold for each word could be based on the product of a measure of the content-value of the word and the acoustic constraint of the word; Group I having the strictest acceptance threshold and Group IV having the most lenient acceptance threshold.

A speech system which searches the word hypotheses to find syntactically legal sequences of words should assume the existence of a Group I word any time it needs one to extend a sequence. Later, these words can be verified for the best sequences of words.

The important thing to note is that the word hypothesizer can be controlled by preset thresholds or dynamically by a speech system to hypothesize those words most

useful to the task of understanding speech.

6.4 Performance Comparison with other Word Hypothesizers

6.4.1 POMOW-Wizard

As described in Section 1.3, the POMOW word hypothesizer of Hearsay-II [Smith - 1976] passes an average of 90 word hypotheses for each utterance word to the Wizard word verifier [McKeown - 1977][4] to be rated. We will treat the combination of these two knowledge sources as a single word hypothesizer for a comparison with Noah.

	POMOW-Wizard	Noah
Performance:		
Word Accuracy:	64%	73% of Words
Avg. Number of Word Hypotheses per Utterance Word:	72	20 Hypotheses
Average Rank of Correct Words:	5.6	2.9 Rank
Size:		
Storage for 1000 Words:	37K	11K (36 bits)
Program and other Storage:	46K	76K
Total:	83K	87K
Computation Costs:		
Number of Million Instructions Per Second of Speech:	28	3.1 MIPSS
Times real-time for a PDP-KL10:	22	2.7 x Real Time

Table 6.3: Comparison of POMOW-Wizard and Noah for the 1000-Word Vocabulary

Table 6.3 compares the word hypothesizers on three dimensions. Noah, using about the same amount of space and about one-tenth the computation time, performs much better than POMOW-Wizard[5]

---

4 Wizard is, in effect, a miniature Harpy system for individual words, rather than complete utterances. Its knowledge is the 1000-word segment-label network dictionary of the Harpy system, which was built manually by looking at samples of segmented and labeled speech.

5 The size comparison for the program parts of the hypothesizers is a little uncertain. Both hypothesizers include debugging and analysis code. The code for Wizard (13K

The improvement in speed for Noah is attributed to a) its ability to eliminate groups of poorly matching word hypotheses by rejecting hypotheses at the low levels and b) the efficiency of a tree search compared to the various search strategies found in POMOW-Wizard. One-third of POMOW-Wizard's time is spent in POMOW generating the 90 word hypotheses per utterance word; two-thirds of the time is spent by Wizard to verify them.

We attribute the improvement in performance of Noah to its greater amounts of speech and segmenter-labeler specific knowledge. The lower word accuracy of 64% for POMOW-Wizard (as opposed to 73% for Noah) is due mainly to POMOW (Wizard rejects only 2% of POMOW's correct word hypotheses). Its poorer average rank of 5.6 (as opposed to 2.9 for Noah) is due to Wizard. As described in Section 1.3.1, POMOW's knowledge is limited to what can be learned from the segment labels for seven equivalence classes of phonemes and stored in a Markov probability model. Wizards knowledge is obtained manually by looking at samples of segmented and labeled speech for each word. Noah was able to automatically learn segment patterns from more samples than was possible for Wizard and store more detail than did POMOW.

6.4.2 Lexical Retrieval Component in the HWIM System

If Noah is restricted to hypothesizing only the best-rated 15 across the whole utterance, its performance can be compared to that of the Lexical Retrieval Component of the HWIM System [Klovstad - 1976]. The Lexical Retrieval Component "scans" each utterance (using a 1097-word vocabulary) to find the best matching words rated better than a set threshold; up to 15 words are found.

The comparison is somewhat artificial because the systems are using different lower level acoustic processors. Noah as using a segmenter-labeler; the Lexical Retrieval Component is using an "Acoustic Phonetic Recognizer" [Woods, et al. - 1976]. The APR hypothesizes phones (a higher level than the segment level) and has a best-rated phone label accuracy about 8% higher than the best-rated segment label accuracy for the segmenter-labeler which Noah uses (52% compared to 44%)[6]. However, an error for a phone label is more costly than an error for a segment label, since the phone label attempts to give more information. In addition, the word hypothesizers are using vocabularies which differ in content as well as in size (1011 words for Noah versus 1097 words for the HWIM system) and different test utterances.

---

of the 46K total) also includes code not used for POMOW hypotheses; this code is used for verifying words hypothesized by the syntactic and semantic knowledge source of the system. (Noah was not intended to replace this task of the verifier.)

6 These values have been normalize to account for the difference of 73 phone labels used by the APR and 98 segment labels used by the segmenter-labeler of the Hearsay-II system.

The following two tables compare the performances of the two systems:

	Noah	HWIM Lexical Retrieval Component
Avg. No. of Words per Sentence:	6.71	6.20
Avg. No. of Word Hypotheses:	15.00	12.30
Avg. No. of Correct Words:	2.86	2.17
Avg. No. of Incorrect Words:	12.14	10.13
Avg. Ratio of Correct to Incorrect Words:	.2356	.2142

Table 6.4: Correct versus Incorrect Word Hypotheses

Rank:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	>15
Noah %:	69	82	86	88	92	94	95	96	96	96	96	97	98	98	98	100
HWIM LRC %:	58	74	80	84	85	86	88	88	90	91	92	92	92	92	92	100

Table 6.5: Rank Distribution of Best Correct Word Hypothesis

According to both measurements -- the ratio of correct to incorrect words and the rank distribution of the best correct word -- Noah performs somewhat better than the Lexical Retrieval Component of the HWIM system. However, we should keep in mind the difficulties in comparing these systems. No data is available on the speed and size of the lexical retrieval component.

AD-A049 287 CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER --ETC F/G 17/2
WORD HYPOTHESIZATION FOR LARGE-VOCABULARY SPEECH UNDERSTANDING --ETC(U)
OCT 77 A R SMITH F44620-73-C-0074

UNCLASSIFIED AFOSR-TR-78-0005 NL

2 OF 2
AD
A049287

END
DATE
FILMED
3-78
DDC

Chapter 7 Summary and Conclusions

This final chapter summarizes the thesis and draws some conclusions before giving the contributions of the thesis. The final section suggests improvements for Noah and points to future work.

7.1 Summary

This thesis describes research directed toward the development of a general English speech understanding system. In particular, the thesis presents the design and performance of a bottom-up word hypothesizer (Noah) capable of handling very large vocabularies. The design of Noah is based on a hierarchy-tree structure. Speech is represented at four levels of a hierarchy. A tree maps the representation of speech at one level to the representation of speech at the next higher level by a tree. Since this design of Noah has been summarized in Chapter 1 (Section 1.4: Overview of Noah), we restrict the summary here to Noah's performance and runtime characteristics.

cont on p 90

7.1.1 Performance

The word hypothesizer was trained on 174 utterances and tested on 105 new utterances (705 words) for 7 different vocabulary sizes ranging from 500 words to 19,000 words. The performance[1] for these vocabularies ranged from a word accuracy of 73% at an average rank of 2.6 for the 500-word vocabulary to a word accuracy of 58% at an average rank of 5.8 for the 19,000-word vocabulary. The rank of a word hypothesis was limited to be less than or equal to 20 (i.e., a maximum of 20 hypotheses per word were allowed). According to the average efficiency measure, which combines word accuracy and rank measures, this performance degrades at approximately a logarithmic rate over the range of vocabulary sizes tested.

An analysis of the effect of training sample size on performance shows that Noah's performance will not improve much more with training if the current method of using arbitrary training utterances is used. We therefore suggested (Section 6.3.2) that selective training is needed, i.e., training for those sylparts for which recognition errors are made.

1 See Section 6.1.1 for an explanation of performance measures.

7.1.2 Runtime Characteristics

The Noah word hypothesizer requires about 87K of 36 bit-words for storage of 1) recognition, debugging, analysis, and statistics code, 2) runtime constants and variables, 3) hypotheses for a 3-second utterance, 4) sylpart training-dependent knowledge (174 training utterances), and 5) vocabulary-dependent knowledge (1000 words). The vocabulary dependent knowledge for 1000 words uses 11K of memory. This increases to 148K (13 times greater) for 19,000 words.

The computation costs begin at 2.4 MIPSS (million of instructions per second of speech) for the 500-word vocabulary and increases at a logarithmic rate to 6.6 MIPSS for the 19,000-word vocabulary. (This is an increase of about .75 MIPPS per doubling of the vocabulary size). In terms of processing time for a PDP-KL10 processer (the 1.3 MIPS machine used for the tests), the time to hypothesize the words for an utterance ranges from 3.1 to 8.6 times the time it takes to speak the utterance, over the same range of vocabulary sizes. If the high degree of parallelism permitted by the recognition algorithm were exploited, these times could be greatly reduced.

(cont'd p 147 3/4)

7.2 Conclusions

The author concludes

The major conclusion from the results of the thesis is that bottom-up word hypothesization is not greatly effected by the size of the vocabulary. We were pleasantly surprised that the effect of vocabulary size on performance and on computation costs would be approximately according to the logarithmic of the vocabulary size. This result is very pleasing in that it suggests that, with improvements in the word hypothesizer and the segmenter-labeler, speech understanding systems for general English can obtain a great amount of constraint from the acoustics alone. Since the main thrust of this thesis was not in building an optimal word hypothesizer but in building one which could handle large vocabularies, many possible performance improvements were set aside because of the time and effort needed to implement and test them. These are suggested in the final section. It is also expected that better segmenter-labelers will come about; when they do, it will be easy to adapt Noah to them and thus improve its performance.

He was

One thing that cannot be concluded from the thesis is that building a speech understanding systems for general English will now be easy. It is not known how a speech system reacts to a bottom-up word hypothesizer with the rate of performance degradation given here. Also, it is not known how such a system reacts to an increase in the complexity of the grammar beyond the narrow ranges tested so far. Of course, any improvement in the word hypothesizer will make the job easier.

The thesis has shown that for word hypothesization it is possible to handle many of the coarticulation problems at a low level in the recognition algorithm. This permits storing only a base pronunciation for words, saving storage and saving effort in acquiring acoustic descriptions of words. However, it is quite possible that for word verification, more-detailed descriptions will have to be stored, either at the word level or at the syllable level.

7.3 Contributions

The thesis has contributed in the following ways:

> The thesis is a step toward speech understanding systems for general English. It has shown that increasing the vocabulary size for a particular bottom-up word hypothesizer, decreases its performance and increases its computation costs approximately according to the logarithmic of the vocabulary size. This in turn has given a feel for the complexity of the "word sound similarity space" for English.

> The thesis presents the design of a bottom-up word hypothesizer which performs better than the POMOW word-hypothesizer / Wizard word-verifier of the Hearsay-II system and the lexical retrieval component of the HWIM system (the only other known bottom-up word hypothesizers). The Noah word hypothesizer has a word accuracy of 73% with an average rank of 2.9 for the correct hypotheses when using a 1000-word vocabulary. This compares to a word accuracy of 65% at an average rank of 4.5 for POMOW/Wizard. Also, Noah runs almost an order of magnitude faster than POMOW/Wizard. Noah's best rated word hypothesis for an utterance is correct 69% of the time. This compares favorably to a value of 58% for the lexical retrieval component of the HWIM system.

> The thesis demonstrates a solution to the problem of knowledge acquisition for AI knowledge-based systems. The solution is to separate the knowledge into a) a priori knowledge: general knowledge which is easily acquired and b) learned knowledge which completes the general knowledge, is acquired by training the system, and is specific to the particular conditions under which the system operates. For word hypothesization this solution has taken the form of a) acquiring base word pronunciations from a pronunciation dictionary as the a priori knowledge and b) automatically learning segment-label patterns of a particular segmenter-labeler for subparts of the word pronunciations (i.e., sylparts). Thus, the word hypothesizer is able to acquire the knowledge for 19,000 words, but at the same time remain free from ties to a particular segmenter-labeler.

> A hierarchy-tree structure of knowledge representation was presented that

gives a way of combining the advantages of both a hierarchy structure and a tree structure for reducing the costs of a bottom-up recognition algorithm. The multiple levels of the structure prevent a potential combinatoric explosion of alternative hypotheses by permitting control of the hypotheses at each level. In addition the structure provides a framework for a) storing a priori knowledge and learned knowledge separately and b) using both types of knowledge jointly.

> The thesis demonstrates a method of learning the context of a pattern and using that context during recognition to constrain the possible interpretations of the pattern. This context learning handles some of the coarticulation problems present at the sylpart level of speech by making the interpretation of a segment-label pattern (i.e., the sylpart it represents) dependent in part on the similarity of a) the context (i.e., left and right adjacent segment labels) of the segment-label pattern in the speech being recognized, and b) context previously learned for the segment-label pattern.

> Two measures (average efficiency, and the confusion of hypotheses) were given which are useful for understanding systems which use the hypothesis-and-test paradigm. The average efficiency measure gives a way of reducing to one number the inter-related measures of a) the number of correct hypotheses and b) their ranking amongst incorrect hypotheses. This makes monitoring of the system's performance for design changes and different test conditions easier. The confusion measure for hypotheses shows how the competition of hypotheses changes in different parts of the system as information is used to create new hypotheses from old hypotheses.

7.4 Other Applications

In this section we speculate on other possible applications of the research. In particular, we will look briefly at using the knowledge and knowledge representation of Noah for investigating what has been called the "word sound similarity space". The hierarchy-tree representation of knowledge is suggested for use in image understanding.

7.4.1 Analysis of the Word Sound Similarity Space

One can imagine an abstract multidimensional space in which the sound of a word is represented by a point in the space (or perhaps by a set of points to account for the many ways the word might occur in speech) and the similarity between the sound of two words is represented by the distance between their corresponding points as measured by some metric. We call this the word sound similarity space. The knowledge and knowledge representation of Noah is one possible model of this space. For the

thesis, this model was used to identify unknown points in the space (defined by segment patterns) by finding the "closest" prelabeled points (i.e, the words of the vocabulary). It is possible to use the same model to investigate the distribution of words in the space. For example, the following kinds of questions could be answered for different vocabularies: What are the regions of greatest density -- that is, what words have the most competition? If a set of words is added to the vocabulary what is the expected change in the performance of the word hypothesizer? How does the word density of one vocabulary compare with another vocabulary?[2]

To answer these questions, the hierarchy-tree representation can be used to compute the "distance" between words. This distance can be defined by recursively finding the distance between the parts of the words which are stored in the levels of the representation: The distance between two words is based on the distance between their corresponding syllables; the distance between syllables is based on the distance between the corresponding sylparts; the distance between sylparts is based on an average distance between their several segment patterns; and the distance between segment patterns is based on an experimentally-derived confusion matrix for segment labels. The real advantage of Noah's particular model of the word space becomes apparent when one wants to find, in parallel, the closest words to a particular word (or perhaps a sequence of words). This is done in two steps: the word is expanded recursively into its syllables and sylparts and then into the segment patterns for each sylpart. These segment patterns are then used as input to Noah. By forcing Noah to use a predetermined segment label-to-label distance metric rather than individual rated segment labels, it can "hypothesize" words based on the segment patterns of the particular word to find its closest words. By using different "input" words the above questions can be answered quickly.

7.4.2 Image Understanding

Before speculating on the applications of the hierarchy-tree representation to the domain of image understanding, we will outline some the concepts present in the representation (these are not necessarily unique to this representation). These concepts are: a hierarchy of levels gives levels of abstraction to the knowledge; each level is defined by a lexicon of units; a tree structure between adjacent levels ties the levels together by storing the patterns of units at the lower level to define a higher level unit; a tree permits storing compactly the knowledge which is peculiar to a pair of levels; and finally, contextual information is stored to constrain better the interpretation of the units at one level for the next higher level.

Image understanding also fits this model. Vision is commonly represented by a

2 This is a question that [Goodman - 1976] investigated for smaller vocabularies.

hierarchy of levels of abstraction ([McKeown & Reddy - 1977] give 6 possible levels: pixel, patch, region, object, cluster, and scene). A tree could be used to store compactly the patterns of units at one level to define the units at the next higher level. The major difference between these trees and those found in Noah is that they would not store time adjacent units but positionally adjacent units. Techniques would have to be developed to linearize the vision patterns for storage. The lower-level trees could be used more for detailed positional knowledge; the higher-level trees could contain a more semantic type of knowledge. In all of the trees, contextual information could be used to constrain the interpretations of the patterns (positional context at the lower levels, semantic context at the higher levels). We believe that other problem domains using the above concepts could also use the hierarchy-tree representation.

7.5 Future Research

The discussion of future research is divided into a section on suggestions for improving Noah and a section on research for intergrating Noah into a total speech system.

7.5.1 Suggested Improvements for Noah

The suggested improvements are given in order of increasing probable gain in performance. This generally corresponds to an increasing amount of work.

7.5.1.1 Tuning the System

Though tuning can be a never-ending task, it is expected that better performance can be obtained (particularly for multisyllabic words) by tuning the current thresholds and parameters which control the rating of hypotheses at each level. These thresholds and parameters control context scoring, weight penalties, normalization for the length of a pattern, and acceptance of hypotheses at each level.

7.5.1.2 Selective Training

Learning the segment patterns and context for those sylparts causing errors is suggested for increasing the effectiveness of training. One way to do this is to incorporate in the word hypothesizer a method of automatically training on the segment patterns for those sylparts which cause a word to be missed. Much of the work has already been done for this in the form of an analysis program which shows why a word was missed or why it was rated poorly. The names and positions of the correct words could be obtained either from interaction with the user or from feedback from the total speech system when it manages to recognize the utterance in spite of some missing correct bottom-up word hypotheses.

7.5.1.3 Additional Information

More information can be used by Noah to improve its performance. We suggest four types of information here: stress, context, duration and syllable boundaries. Each of these would require considerable research to design, test, and adjust. Another problem is that an increase in the amount of information used by Noah causes an even greater increase in the amount of training needed.

Syllable stress information could be used in several ways: 1) as a further test for multisyllabic words -- the stress pattern of a word would have to match the stress pattern of the utterance, 2) as a way of reducing erroneous function word hypotheses and 3) as part of the vowel recognition process -- a schwa should probably not be hypothesized in a stressed location.

The context used by Noah for recognizing sylparts is small. This could be increased to more segments as required by particular sylparts or to other types of information, such as the stress of surrounding syllables. The difficulty with learning and using context is that the system or designer must determine when the context is really effecting the patterns for the sylparts.

The duration of the segments in each segment pattern was once used in the Noah recognition algorithm without success. Duration information may still be useful if it is used for the total segment pattern. Duration information could also be used in conjunction with amplitude information to determine syllable stress.

It is firmly believed that detection of syllable boundaries would greatly reduce the number of incorrect syllable hypotheses and in turn, incorrect word hypotheses. An example of the confusion resulting from lack of syllable boundaries was seen in Figure 6.6. Mermelstein [Mermelstein - 1975] has demonstrated the ability to recognize syllable boundaries with an error rate of 6.9% syllables boundaries missed and 2.6% extra syllables boundaries found. It seems that this performance like this would greatly increase Noah's performance for large vocabularies.

7.5.2 Speech System Integration for Noah

7.5.2.1 Performance within a Particular System

More work needs to be done in evaluating Noah within the constraints of particular speech systems. For example, suppose a system is not very good at dealing with missing correct word hypotheses, but is able to handle many hypotheses efficiently and effectively. The performance evaluation of Noah relevant to such a system is the percent of correct word hypotheses at a constant average rank as the vocabulary increases, without much regard to the total number of hypotheses. The measures of performance given in the thesis are not weighted in this direction. In addition, Noah

should be analyzed in terms of cost-effectiveness for the system using it.

7.5.2.2 System Control

The performance of Noah can be improved if it is controlled intelligently by the speech system using it. Any a priori knowledge that can be used to constrain the word hypothesizer effectively increases its performance. Three types of constraints are suggested here. Each of them can be applied over the whole utterance or in selected parts of the utterance.

As discussed in Section 6.3.7, the word hypothesizer can be controlled to hypothesize words with specific characteristics. For example, function words can be "tuned out". The overall needs of the particular speech system or its needs at particular parts of the utterance can be adjusted for.

Word hypotheses could also be biased *a priori* by the expected topic of the utterance. For example, as in these tests, if the expected topic was Artificial Intelligence articles, the words of the 1000-word Hearsay-II dictionary could be biased over the other words in the 19,000-word dictionary. This biasing can be done by penalizing those words not included by the topic. Only penalizing of words should be done and not elimination, to permit the speaker to change topics.

Finally, words at a particluar location in the utterance could be selected by the speech system for hypothesization according to a particular part of speech and topic (e.g., color adjective, person's name, etc.). Generally, this is done by word verifiers for speech systems, but if the set of words (such as "all nouns") is very large the word hypothesizer should be called. Since the vocabulary size is effectively reduced by these constraints, the thresholds of the word hypothesizer could be relaxed to avoid missing the correct word hypothesis.

7.5.3 Great Expectations

It is expected that the performance of bottom-up word hypothesization can be improved to a word accuracy of 80% at an average rank of 3 with an average of 10 competing hypotheses for each utterance word, using a 20,000-word vocabulary. This improvement would be made by including the above suggestions and by improvements in segmenting and labeling. Though in some sense these numbers are "out of the blue", we have seen that the performance of the Noah word hypothesizer equals the performance of the POMOW word hypothesizer and the Wizard word verifier of the Hearsay-II speech system using a vocabulary almost 8 times smaller. We estimate that the above numbers represent the same magnitude of performance increase over Noah and could be achieved with about the same amount of effort (about two man-years). If the performance of such an hypothesizer were effected by large vocabularies in the

same way as Noah (and there is no reason to think otherwise), it would have a word hypothesization accuracy of 73% at an average rank of 4.5 for a 100,000-word vocabulary.

References

Bahl, L. R., et. al. Preliminary results on the performance of a system for the automatic recognition of continuous speech. *1976 IEEE Inter. Conf. on Acoustics, Speech and Signal Processing*, Philadelphia, Apr., 1976, 425-429.

Baker, J. K. Stochastic modeling as a means of automatic speech recognition. Tech. Report, CMUCSD, 1975. Ph.D. Dissertation -- The Dragon system.

CMU Computer Science Speech Group. (1976) Working papers in speech recognition - IV - the Hearsay-II system. Tech. Report, CMUCSD, 1976.

CMU Computer Science Speech Group. (1977) Summary of the CMU five-year ARPA effort in speech understanding research. Tech. Report, CMUCSD, 1977.

Cole, A. The ARPA-SUR phonological rules: summary and index. ARPA Speech Understanding Research Note No. 136., May 1974., (Cole is at) the Computational Speech and Language Group, Electronics Research Laboratory, University of California at Berkeley.

Erman, L. D. A functional description of the Hearsay-II system. *Proc. 1977 IEEE Inter. Conf. on ASSP*, Hartford, CT, May, 1977, 799-802.

Feigenbaum, E. The Art of Artificial Intelligence: I. Themes and case studies of knowledge engineering. *Proc. IJCAJ-77*, Mass., Aug., 1977, 1014-1029.

Forgie, J. W. Overview of the Lincoln System. *IEEE Symposium on Speech Recognition*, Carnegie-Mellon University, 1974, 27.

Goldberg, H. Performance of the Hearsay-II segmenter-Labeler., Private communication, 1977.

Goodman, G. Analysis of languages for man-machine voice communication. Tech. Report, CMUCSD, May, 1976. (Ph.D. Dissertation, Comp. Sci. Dept., Stanford University)

Hayes-Roth, F. and Mostow, D. J. An automatically compilable recognition network for structured patterns. *Proc. IJCAJ-75*, Tbilisi, USSR, Aug., 1975. Also appeared in [CMU Computer Science Speech Group - 1976].

Hayes-Roth, F., Mostow, D. J., and Fox, M. Understanding speech in the Hearsay-II system. In *Natural Language Communication with Computers*. (Bloc, L., Ed.) Springer-Verlag, Berlin, 1977, (in press).

Itakura, F. Minimum prediction residual principle applied to speech recognition, *1975 IEEE Trans. ASSP-23*, 67-72.

Klovstad, J. W. Probabilistic lexical retreval component with embedded phonological word boundary rules. Technical Report In Woods, et al., *Speech Understanding Systems Technical Progress Report No.* 6, Bolt, Beranek and Newman Inc., Apr., 1976, 68-108. Also a PH.D. Dissertation, to be published.

Knuth, D. (1968) *The Art of Computer Programming: Vol. 1, Fundamental Algorithms*, Addison-Wesley, Menlo Park, Ca., 1968.

Knuth, D. (1973) *The Art of Computer Programming: Vol. 3, Sorting and Searching*, Addison-Wesley, Menlo Park, Ca., 1973.

Lowerre, B. T. The Harpy speech recognition system. Tech. Report, CMUCSD, 1976. Ph.D. Dissertation.

McKeown, D. M. and Reddy, D. R. (1977) A hierarchical symbolic representation for an image database, *Proceedings of IEEE Workshop on Picture Data Description and Management*, Chicago, Ill., Apr., 1977.

McKeown, D. M. Word verification in the Hearsay-II speech understanding system. *Proc. 1977 IEEE Inter. Conf. on Acoustics, Speech and Signal Processing*, Hartford, CT, May, 1977, 795-798.

Mermelstein, P. Automatic segmentation of speech into syllabic units. *Journal of the Acoustic Society of America*, 58., 1975, 880-883.

Miller, G. and S. Isard Some perceptual consequences of linquistic rules. *Journal of Verbal Learning and Verbal Behavior*, Vol. 2, 1963, 217-228.

Newell, et al. Speech understanding systems: final report of a study group. North Holland, 1973. (Originally appeared in 1971).

Newell, A. A tutorial on speech understanding systems. In *Speech Recognition: Invited Papers of the IEEE Symp.*. (Reddy, D. R., Ed.) Academic Press, New York, NY, 1975, 3-54.

Olney, J., and D. Ramsey From machine dictionaries to a lexicon tester: progress, plans and an offer. *Computer Studies in the Humanities and Verbal Behavior*, Vol. 3, No. 4, Nov., 1972, 213-220.

Reddy, D. R. An approach to computer speech recognition by direct analysis of the speech wave. Tech. Report, Stanford University, AI Memo 43, Stanford, CA, 1966. Ph.D. Dissertation.

Reddy, D. R. and Vicens, P. J. A procedure for segmentation of connected speech. *J. Audio Engr. Soc. 16* Apr., 1968 404-412.

Reddy, D. R., Erman, L. D. and Neely, R. B. (1972) A mechanistic model of speech perception. *Proc. 1972 IEEE Conf. Speech Communication and Processing*, Newton, MA, Apr., 1972, 334-337.

Reddy, D. R., L. D. Erman, and R. B. Neely (1973) The HEARSAY speech understanding system: an example of the recognition process. *Proc. 3rd Inter. Joint Conf. on Artificial Intel.*, Stanford, Ca., 1973, 185-193.

Rubin, F., Experiments in text file compression. *Comm. of the ACM*, Vol. 19, Nov., 1976, 617-623.

Shannon, C. E., A mathematical theory of communication, *Bell System Technical Journal*, 27, 1948.

Shockey, L. and Adam, C. The phonetic component of the Hearsay-II speech understanding system. In [CMU Computer Science Speech Group - 1976].

Sivertsen, E., Segment inventories for speech synthesis, based on University of Michigan Speech Research Laboratory Report No. 5., 1961.

Smith, A. R. Word hypothesization in the Hearsay.-II speech system. *Proc. 1976 IEEE Inter. Conf. on ASSP*, Philadelphia, PA, Apr., 1976, 549-552. Also in [CMU Computer Science Speech Group - 1976].

Vicens, P. J. Aspects of speech recognition by computer. Tech. Report, Stanford University, AI Memo 85, Stanford, CA, 1969. Ph.D. Dissertation.

Woods, W. A. Transition network grammars for natural language analysis, *Comm. of the ACM*, Vol. 13, Oct., 1970, 591-606.

Woods, W., et. al. Speech understanding systems: final report. Bolt, Beranek and Newman Inc., Oct., 1976.

Appendix A: "ARPABET" Computer Phonetic Representation

Phoneme	Computer Representation	Example	//	Phoneme	Computer Representation	Example
i	IY	beat		n	N	net
I	IH	bit		ŋ	NX	sing
e	EY	bait		p	P	pet
ɛ	EH	bet		t	T	ten
æ	AE	bat		k	K	kit
a	AA	bomb		b	B	bet
ʌ	AH	but		d	D	debt
ɔ	AO	bought		g	G	get
o	OW	boat		h	HH	hat
U	UH	book		f	F	fat
u	UW	boot		θ	TH	thing
ə	AX	about		s	S	sat
ɝ	ER	bird		š	SH	shut
au	AW	down		v	V	vat
aI	AY	buy		ð	DH	that
ɔI	OY	boy		z	Z	zoo
y	Y	you		ž	ZH	azure
w	W	wit		syl l	EL	battle
r	R	rent		syl m	EM	bottom
l	L	let		syl n	EN	button
m	M	met				

Appendix B: Lexicons

This appendix includes the sylpart lexicons, a list of the segment class labels, and the 2000-word lexicon.

Sylpart Lexicons

The number after each entry gives the number of training samples found for the sylpart in the 174 utterance training set.

VOWEL LEXICON

IY 118	IH 264	EY 68	EH 108	AE 107
AA77	AO 19	OW 38	UH 3	UW 47
IX 8	AX 295	ER 86	AW 27	AY 71
OY 8	EL 38	EN 5	EM 14	

VOWEL SEQUENCE LEXICON

EH L 4	OW R 3	Y UW 8	AX AO 6	OW L 3
AA R 23	AO R 9	R IH 17	EH R 3	HH ER AX 1
IY EY 4	R OW 3	IY AA 4	AX N 2	L IY 6
AX AE 4	UW EH 6	IH L 5	IY AE 5	R EY 3
IY R IY 3	EH R IH 1	AE L 1	UH R EL 1	IY WH EH 1
UW IH 1	IY AX 9	AA L AX 2	ER IH 3	AE M IH 1
W AX 3	R AE 7	EL AX 4	ER AX 5	R AX 9
AY AX 1	W UH 1	Y UW ER 1	AE R 5	EH L AX 2
UW AO L 1	W IH 2	L OW 1	UW AX 1	AA R AX 1
IH N 1	N AY 1	R EH 7	A OL 5	ER R EH 1
IY R EM 1	IY EH 6	IY AA R 1	L EY IH 1	IH R 4
UW Y UW HH AE 1		AY EM 1	Y ER 1	UW ER 1
R IY 3	AA R EH 1	AX OW 1	AE R AO 2	AA L 1
ER L AY 1	L AE 1	AE R EH 3	L IH 3	UW HH AE 1
WH IH 1	HH IH R 1	DH AX 1	AX D AX 1	ER IY 2
EY AY 2	IH L IY 1	UH R IH 2	AX W EH R 1	UH L AY 1
EH R IY 1	R EM 8	AE R AX 1	EH T ER 1	Y UW EL 1
DH AX AE 1	R EL 1	AX EY 3	AX R AX W 1	EY IH 1
AY R AX 1	UH L 1	OW R IY 1	IY D IY 1	IY R 1
AO R IY 1	AA L IH 2	R EY IH 1	R AA 1	AE R IY 1
AE L AX 1	IY IH 1	L EM 1		

ONSET LEXICON

P 91	T 88	K 64	B 5	D 85
G 24	F 48	TH 18	S 97	SH 24
ZH 2	V 17	DH 87	Z 10	M 82
N 44	L 47	R 29	HH 43	WH 53
Y 9	W 23	M Y 2	N Y 12	P L 9
P R 10	P Y 3	T SH 22	T R 13	T W 1
T Y 1	K L 3	K R 1	K Y 1	K W 4
B L 2	B R 2	B Y 2	D ZH 13	D R 3

D Y 4	G L 2	G R 5	G W 5	HH Y 2
F R 2	TH R 2	S P 4	S T 24	S K 13
S M 1	S N 1	S L 2	V Y 1	S P L 1
S P R 1	S T R 11	S K R 1		

CODA LEKICON

R 19	L 13	M 22	N 25	NX 53
P 14	T 133	K 33	B 26	D 35
G 6	F 18	TH 1	S 42	SH 19
V 82	DH 3	Z 121	ZH 1	R M 1
R N 1	L P 1	M P 1	S P 1	R T 16
N T 14	K T 6	S T 26	SH T 4	NX K 3
S K 2	R D 2	L D 1	N D 13	V D 1
Z D 1	L F 1	R TH 1	N TH 2	P TH 1
R S 1	L S 1	N S 4	P S 1	T S 7
K S 18	F S 1	T SH 17	L V 1	R Z 1
L Z 1	M Z 3	N Z 14	N XZ 2	B Z 1
D Z 2	V Z 2	D ZH 9	R N T 2	N S T 2
K S T 4	R T S 2	N T S 2	K T S 12	S T S 1
L K S 1	NX K S 1	R T SH 2	N T SH 2	R M Z 1
N D Z 2	L V Z 1	R D ZH 1		

Segment Class Labels

ASF - aspiration or fricative
ASH - aspiration or high energy fricative
ASL - aspiration or low energy fricative
ASP - aspiration
BAR - voice-bar
DCN - dip position consonant
FLB - voice-bar or flap
FLP - flap
FRC - fricative
FRU - unvoiced fricative
FRV - voiced fricative
FSI - fricative or silence
FVB - final voice-bar or nasal
GLQ - liquid or glide
HFR - high energy fricative
HHY - HH or Y
HUF - high energy unvoiced fricative
HVF - high energy voiced fricative
LFR - low energy fricative
LUF - low energy unfoiced fricative
LVF - low energy voiced fricative
NAS - nasal
NGL - nasal, glide, or liquid
NVF - nasal or voiced fricative
PLS - low energy speech sound
SIL - silence
SPL - sonorant voicing minimum
SPP - sonorant voicing peak
TCN - tail position consonant
FRU - unvoiced fricative
VAA - AA-like vowel
VBK - back vowel
VCN - non-nuclear vowel or resononant consonant
VFT - front vowel

VIV - IY-like vowel
VLW - low vowel
VMD - mid vowel
VSW - schwa-like vowel
VUW - UW-like vowel

The 2000-Word Lexicon

The code following each word indicates the smallest vocabulary in which the word is found: (T) the word occurs in the test utterances, (5) 500-word vocabulary, (1) 1000-word vocabulary, and (2) 2000-word vocabulary.

A /T	ABDICATION/2	ABJURE/2	ABOMINATE/2	ABOUT/T	ABSENT/2
ABSTRACT/T	ABSTRACTION/1	ABSTRACTS/T	ABUT/2	ACCEPTANCE/2	ACCOMPLICE/2
ACCUSATION/2	ACIDITY/2	ACL/T	ACM/1	ACQUISITION/1	ACRIDITY/2
ACTIONS/1	ACTIVE/5	ACTUAL/2	ACYCLIC/1	ADAPTATION/5	ADAPTIVE/5
ADDICT/2	ADDITION/5	ADDRESS/5	ADDRESSES/T	ADJOIN/2	ADMIRER/2
ADRENAL/2	ADVERSARY/2	ADVISING/5	AERATION/2	AESTHETICS/5	AFAR/2
AFFILIATION/5	AFFILIATIONS/5	AFFLUENCE/2	AFTER/5	AFTERBIRTH/2	AGGRANDIZE/2
AGO/2	AI/T	AILMENT/2	ALBINO/2	ALEMBIC/2	ALGEBRAIC/5
ALGOL/5	ALGORITHM/T	ALGORITHMIC/5	ALIMENTARY/2	ALL/T	ALL-OR-NONE/5
ALLEN/T	ALLOCATION/2	ALONE/2	ALSO/T	ALTERATION/2	ALWAYS/T
AM/T	AMAH/2	AMBERGRIS/2	AMENDS/2	AMMONITE/2	AMONG/1
AMPHORA/2	AN/T	ANAL/2	ANALOGY/5	ANALYSIS/T	ANALYZER/1
ANATOMY/2	AND/T	ANESTHETIST/2	ANIMAL/2	ANN/1	ANNIHILATE/2
ANOMALOUS/2	ANOTHER/5	ANSWER/T	ANSWERING/1	ANTENNA/2	ANTHONY/1
ANTICHRIST/2	ANTIQUITY/2	ANY/T	ANYONE/5	ANYTHING/T	ANYWHERE/T
APARTHEID/2	APIECE/2	APOTHEOSIS/2	APPEAR/1	APPEARED/T	APPENDIX/2
APPLICATION/1	APPOINTMENT/2	APPRENTICE/1	APPROACH/1	APPROXIMATION/2	APRIL/5
ARABESQUE/2	ARBIB/5	ARCADE/2	ARCHITECTURE/2	ARE/T	AREA/T
AREAS/T	AREN'T/5	ARMISTICE/2	ARPA/5	ARREST/2	ART/T
ARTICLE/T	ARTICLES/T	ARTICULAR/2	ARTIFICIAL/T	ARTS/1	ASCEND/2
ASIMOV/1	ASININITY/2	ASK/T	ASSAIL/2	ASSEMBLY/T	ASSERTIONS/1
ASSIMILATE/2	ASSIMILATION/T	ASSOCIATION/T	ASSOCIATIVE/T	ASTHMATIC/2	ASTROPHYSICS/2
AT/T	ATOMIC/2	ATTAINABILITY/2	ATTENTION/5	ATTRIBUTABLE/2	AUDIO/2
AUGMENTED/T	AUGUST/1	AURAR/2	AUTHOR/1	AUTHORITATIVE/2	AUTHORS/T
AUTOMATED/T	AUTOMATIC/T	AUTOMATION/T	AUTOMOTIVE/2	AVAILABLE/5	AVERMENT/2
AVOUCH/2	AWARD/1	AWN/2	AXIOMATIC/T	AXIOMS/T	AZRIEL/T
BABBLE/2	BACKGAMMON/T	BACTERIOLOGY/2	BAIL/2	BALDRIC/2	BALSAM/2
BANDSTAND/2	BANERJI/1	BANK/1	BANTER/2	BARON/2	BARROW/1
BARTENDER/2	BASE/1	BASEBALL/1	BASED/T	BASES/1	BASSO/2
BATES/1	BATON/2	BAY/T	BAZOOKA/2	BEAST/2	BECKON/2
BEEF/2	BEEN/5	BEFORE/5	BEGONIA/2	BEHAVIOR/1	BELIEF/T
BELIKE/2	BENEATH/2	BENZOL/2	BERKELEY/T	BERLINER/5	BERNARD/1
BERT/1	BEST/2	BETWEEN/1	BEZOAR/2	BIDDEN/2	BIG/1
BILL/T	BILLET/2	BINDING/1	BINDINGS/T	BIOGRAPHY/2	BIOMEDICINE/T
BISEXUAL/2	BLAB/2	BLARNEY/2	BLEDSOE/1	BLEMISH/2	BLITHE/2
BLOCK/1	BLOSSOM/2	BLUEPOINT/2	BOATHOUSE/2	BOBROW/5	BODY/2
BONDMAN/2	BONNIE/1	BOOK/5	BOOKS/5	BOOMTOWN/2	BORNE/2
BOUFFANT/2	BOUNDS/5	BOUTONNIERE/2	BRACE/2	BRAIN/T	BRAND/2
BRAZILIAN/2	BREECH/2	BRIG/2	BRISKET/2	BROIL/2	BROTHER/2
BRUCE/5	BRUTALITY/2	BUCHANAN/5	BUDGERIGAR/2	BULGE/2	BUNCH/2
BUREAU/2	BURMESE/2	BUSH/2	BUSINESS/T	BUT/1	BUTTOCKS/2
BY/T	CABIN/2	CACM/5	CADENZA/2	CAI/T	CAJUN/2
CALCULUS/5	CALF/2	CALLOW/2	CAMPUS/2	CAN/T	CANDIDATE/2
CANNONBALL/2	CANTICLE/2	CAPABILITIES/1	CAPACITY/2	CAPRICE/2	CAPUCHIN/2
CAR/1	CARBONIFEROUS/2	CARFARE/2	CARL/1	CARNIVORA/2	CARRY/2

CARTOGRAPHY/T CASE/T CASHIER/2 CASTIGATION/2 CATALPA/2 CATERCORNER/2
CAUCASIAN/2 CAUSAL/T CAVALCADE/2 CEASE/5 CELANDINE/2 CELL/5
CEMBALO/2 CENTENNIAL/2 CENTURY/2 CERTIFY/2 CHALCEDONY/2 CHANCELLOR/2
CHAPEL/2 CHARGE/2 CHARNIAK/1 CHEAPEN/2 CHECKER/1 CHECKING/T
CHELA/2 CHESS/T CHEVALIER/2 CHIEF/2 CHIN/2 CHITLINS/2
CHOLERIC/2 CHOOSE/5 CHOSEN/2 CHRISTOPHER/1 CHRONOGRAPH/2 CHUCK/1
CHURL/2 CINNABAR/2 CIRCLE/T CIRCUIT/T CIRCUITS/1 CIRCUMLOCUTION/
CITE/T CITED/5 CITES/1 CITY/2 CLAN/2 CLASS/2
CLEANSE/2 CLICHE/2 CLIMBING/5 CLIPBOARD/2 CLOTHIER/2 CLUSTERING/T
CLUTTER/2 CMU/1 COBBLE/2 COCKNEY/2 CODE/1 CODING/5
COEFFICIENT/2 COGNITION/T COGNITIVE/T COGNOMEN/2 COLBY/T COLES/1
COLESLAW/2 COLLEGIAN/2 COLLINS/T COMB/2 COME/T COMET/2
COMMENDATION/2 COMMENTS/1 COMMITTEE/1 COMMON/T COMMOTION/2 COMMUNICATION/1
COMMUNICATIONS/COMPARABLE/2 COMPILATION/2 COMPLEX/T COMPLEXITY/1 COMPONENT/2
COMPONENTS/1 COMPREHENSION/1 COMPUNCTION/2 COMPUTATION/T COMPUTATIONAL/T COMPUTER/T
COMPUTERS/1 COMPUTING/5 CONCEPTUAL/T CONCERN/5 CONCERNED/5 CONCERNING/5
CONCERTINA/2 CONCRETE/2 CONCURRENT/1 CONDOLE/2 CONFEDERACY/2 CONFERENCE/5
CONFERENCES/5 CONFINE/5 CONFORMATION/2 CONGRESSMAN/2 CONNIVE/2 CONSENT/2
CONSIDER/5 CONSIDERED/5 CONSOLIDATE/2 CONSTITUTION/2 CONSTRAINT/T CONSTRUCTING/1
CONSTRUCTION/1 CONSULTANT/1 CONSULTATION/1 CONSULTATIONS/T CONTAIN/T CONTAINED/T
CONTAINS/5 CONTEMN/2 CONTEXT/1 CONTINENT/2 CONTINUOUS/1 CONTRADICT/2
CONTRIVE/2 CONTROL/T CONTROLLED/T CONVENTION/5 CONVENTIONS/5 CONVERGE/2
CONVOKE/2 COOPERATING/1 COOPERATION/1 COOPERATIVE/2 COPULATION/2 COPY/T
COPYING/1 CORPORATE/2 CORRECTNESS/T CORRUGATE/2 COSMIC/2 COTTON/2
COULD/5 COUNTERFEIT/2 COUPLING/2 COVER/2 COYOTE/2 CRANIUM/2
CRAYON/2 CREDO/2 CRESCENT/2 CRIMINOLOGY/2 CROCODILE/2 CROUCH/2
CRUET/2 CRYPTIC/2 CUISINE/2 CUNEIFORM/2 CURIO/2 CURRENT/5
CURVATURE/2 CURVED/1 CUTICLE/2 CYBERNETICS/1 CYCLIC/1 CYCLOTRON/2
DACHA/2 DALLY/2 DANDLE/2 DANNY/5 DATA/5 DATE/T
DATES/1 DAVE/1 DAVID/1 DDT/2 DEB/2 DEBATE/T
DECALCOMANIA/2 DECEMBER/5 DECIPHER/2 DECISION/1 DECOMPOSITION/2 DEDUCE/2
DEDUCTION/T DEDUCTIVE/1 DEFERMENT/2 DEHYDRATE/2 DELIBERATE/2 DELVE/2
DEMAND/5 DEMOCRAT/2 DEN/2 DENOTATIONAL/1 DENTIST/2 DEPORTMENT/2
DEPTH/1 DERELICT/2 DERIVATION/1 DESCRIBE/T DESCRIPTION/1 DESCRIPTIONS/T
DESECRATION/2 DESIGN/1 DESIRE/5 DESPERATE/2 DETACH/2 DETECTION/1
DETRAIN/2 DEVICES/5 DEVITALIZE/2 DIAGNOSE/2 DIAGNOSIS/1 DIALOGUE/1
DIARRHOEA/2 DICK/1 DICTUM/2 DID/T DIDN'T/T DIGIT/2
DIM/2 DIMENSIONAL/1 DINKY/2 DIRECTED/5 DISASSEMBLE/2 DISCOMBOBULATE/
DISCUSS/T DISCUSSED/T DISCUSSES/T DISCUSSING/5 DISEUSE/2 DISLOCATE/2
DISPLAY/1 DISPOSAL/2 DISREPUTE/2 DISSOLUTE/2 DISTRICT/2 DIVERSE/2
DO/T DOCENT/2 DOES/T DOESN'T/5 DOFF/2 DOLOR/2
DOMAIN/1 DON'T/T DONALD/1 DONE/5 DONNYBROOK/2 DOTAGE/2
DOUG/1 DOWAGER/2 DRAFTEE/2 DRAGON/1 DRAGONS/1 DRAWINGS/1
DRAWL/2 DREW/1 DREYFUS/T DRINK/2 DRIVING/1 DROSS/2
DUAL/2 DUENNA/2 DUMMY/2 DUPLICATION/2 DURING/5 DYNAMIC/T
DYNAMITE/2 EACH/5 EARL/1 EARLIEST/5 EARNEST/1 EARTH/2
ECCENTRIC/2 ED/T EDINBURGH/1 EDITORIAL/2 EFFICIENTLY/T EGOISTIC/2
EIGHT/1 EIGHTEEN/5 EIGHTY/5 ELATE/2 ELECTROMAGNET/2 ELECTRONIC/1
ELECTRONICS/1 ELEVEN/5 ELIOE/2 ELLIOT/1 ELUSIVE/2 EMBEZZLE/2
EMEND/2 EMOTION/2 EMU/2 ENCOMPASS/2 ENEMY/2 ENGLISH/5
ENNOBLE/2 ENTERITIS/2 ENTROPY/2 ENVIRONMENT/1 EON/2 EPIPHANY/2
EQUALIZATION/2 EQUIPMENT/2 ERIK/1 ERMAN/T ERNST/1 ERRAND/2
ESCHEW/2 ESPY/2 ETHER/2 EUGENE/1 EUGENICS/2 EVALUATE/2
EVALUATION/1 EVALUATOR/1 EVENTS/1 EVER/T EVERLASTING/2 EVERY/1
EVERYTHING/T EXAMPLE/T EXAMPLES/1 EXCISE/2 EXCURSIVE/2 EXHAUST/2
EXIST/T EXORBITANT/2 EXPERT/1 EXPERTISE/2 EXPLANATION/1 EXPLORER/2
EXPRESSIONS/1 EXPRESSMAN/2 EXTIRPATION/2 EXTRAVAGANZA/2 EYESTRAIN/2 FABLES/1
FACES/1 FACING/2 FACTS/5 FAHLMAN/1 FAILURE/2 FAIRY/1
FALSETTO/2 FANDANGO/2 FARRAGO/2 FASTENER/2 FASTER/T FATUOUS/2
FEATURE/2 FEATURE-DRIVEN/FEBRUARY/5 FEDERAL/1 FEEDBACK/2 FEIGENBAUM/T

FELDMAN/T FELONY/2 FERN/2 FETCH/2 FICHU/2 FICTION/1
FIFTEEN/5 FIFTY/5 FIGHT/2 FIKES/1 FILE/5 FILLING/2
FIND/2 FINISH/5 FINISHED/5 FIRST/5 FIVE/5 FIXATION/2
FOGEY/2 FONDU/2 FOR/T FORBODE/2 FOREIGN/2 FORESTS/1
FORETOKEN/2 FORMAL/5 FORMALIZE/2 FORMATION/T FORTHRIGHT/2 FORTY/5
FOUND/2 FOUR/5 FOURTEEN/5 FRAGRANT/2 FRAME/5 FRAMES/1
FRAUD/2 FRENETIC/2 FRIGHT/2 FROM/T FRONTAGE/2 FRUITION/2
FU/1 FUNCTION/5 FUNCTIONS/1 FUNDAMENTAL/2 FURRIER/2 FUZZY/1
GABFEST/2 GAINSAY/2 GALLIVANT/2 GAME/T GAMES/1 GAMMER/2
GAP/2 GARRISON/2 GARY/1 GASCHNIG/1 GATHER/2 GAZETTE/2
GENEALOGY/2 GENERAL/1 GENERATE/T GENERATION/1 GENUS/2 GEOMETRIC/1
GEORGE/1 GERMINATION/2 GET/T GHOST/2 GIMCRACK/2 GIPS/1
GIVE/T GIVEN/T GLABROUS/2 GLAZE/2 GLOBAL/2 GLUE/2
GM/1 GO/T GO-MOKU/1 GOAL/1 GOALS/1 GOAT/2
GOLDFINCH/2 GOPHER/2 GOUGE/2 GRADIENT/2 GRAIN/5 GRAMMARS/1
GRAMMATICAL/T GRANDIOSE/2 GRAPH?T GRAPHICS/T GRATIFICATION/2 GRAZE/2
GREW/2 GRIP/2 GROMMET/2 GROUNDWATER/2 GRUNT/2 GUILDER/2
GUN/2 GUZZLE/2 HABIT/2 HADES/2 HALVERS/2 HAMBURG/1
HANDOUT/2 HANS/5 HAPHAZARD/2 HAPPEN/1 HARLOT/2 HARRY/5
HAS/T HASN'T/5 HASP/2 HAUTEUR/2 HAVE/T HAVEN'T/5
HAYES-ROTH/1 HE/5 HEADMAN/2 HEARSAY/5 HEATH/2 HEDONISM/2
HELICES/2 HELP/T HELPMATE/2 HENCHMAN/2 HENDRIX/1 HER/5
HERB/5 HERBERT/1 HERDSMAN/2 HERMIT/2 HETEROSTATIC/1 HEURISTIC/5
HEWITT/1 HEWN/2 HIDDEN/2 HILARY/1 HILL/5 HILLOCK/2
HIRSUTE/2 HIS/1 HISTORY/1 HOAK/2 HOLINESS/2 HOLLAND/T
HOMILETIC/2 HONORARIUM/2 HOPE/2 HORRID/2 HOSPITALIZE/2 HOUSEBOAT/2
HOW/T HUGH/1 HUMAN/1 HUMANE/2 HUMMOCK/2 HUNDRED/5
HUNGRY/1 HUNT/1 HURRAH/2 HYBRID/2 HYDROUS/2 HYPHENATION/2
HYPOTHESIS/1 HYSTERIA/2 I/T I'D/T I'M/5 ICY/2
IDIOMATIC/2 IEEE/5 IFIP/5 IGNOMINIOUS/2 IJCAI/T ILLINOIS/1
ILLOGICAL/2 IMAGE/1 IMAGES/1 IMBALANCE/2 IMMATURE/2 IMMODEST/2
IMMUTABLY/2 IMPATIENT/2 IMPERIAL/2 IMPIETY/2 IMPORT/2 IMPRACTICAL/2
IMPROBABLE/2 IMPROVING/1 IMPUTATION/2 IN/T INARTICULATE/2 INCARNADINE/2
INCISE/2 INCOMMODE/2 INCREDIBILITY/2 INCUMBENT/2 INDEFINITE/2 INDICATE/2
INDISPENSABLE/2 INDUCE/2 INDUCTIVE/1 INDUSTRIAL/1 INELIGIBLE/2 INEXACT/1
INFALLIBLE/2 INFERENCE/T INFERENCES/1 INFERENTIAL/1 INFEST/2 INFORMAL/2
INFORMATION/T INGRAIN/2 INHERIT/2 INHERITANCE/1 INJURE/2 INNERMOST/2
INPUT/2 INSANE/1 INSECTIVOROUS/2 INSIGHT/2 INSPIRATION/2 INSTIL/2
INSTITUTE/1 INSULATOR/2 INTEGRATE/2 INTELLIGENCE/T INTELLIGENT/1 INTENSITY/1
INTENTIONS/1 INTERACTION/2 INTERACTIVE/1 INTERESTED/T INTERJECTION/2 INTERMEDIARY/2
INTERNUNCIO/2 INTERPRETABLE/1 INTERPRETIVE/1 INTERRUPTS/1 INTERTIDAL/2 INTERVIEW/1
INTONATION/1 INTROSPECTION/2 INVARIABLE/2 INVARIANCE/1 INVARIANCES/1 INVESTMENT/1
INVIDIOUS/2 INVOCATION/5 INVOLUTE/2 IRASCIBLE/2 IRRECONCILABLE/2 IRRESOLUTE/2
IRV/1 IS/T ISLAND/2 ISN'T/1 ISOMERS/1 ISSAC/5
ISSUE/5 ISSUED/5 ISSUES/5 IT/T ITALIAN/2 ITERATION/1
ITS/5 JACK/1 JACKLEG/2 JAMES/1 JANUARY/5 JAR/2
JEAN/1 JEFFREY/1 JEJUNE/2 JERRY/T JEWEL/2 JITTERS/2
JOHN/T JOLLIFICATION/2 JOSEPH/1 JOURNAL/T JOURNALS/T JOY/2
JUDEA/1 JUDICIAL/1 JUICY/2 JULY/5 JUNE/5 JUPITER/2
KAFFEEKLATSCH/2 KARL/1 KEBOB/2 KEITH/1 KEN/T KEY/5
KEYS/5 KIBBUTZ/2 KILL/5 KIND/1 KINDLE/2 KINDS/5
KING/1 KITCHENWARE/2 KNICKNACK/2 KNOW/5 KNOWLEDGE/5 KNOWN/1
KOREAN/2 KUGEL/1 LABOR/2 LABS/1 LAGNIAPPE/2 LAMBDA/1
LAMINA/2 LANDSMAN/2 LANGUAGE/5 LANGUAGES/5 LAPPET/2 LARGE/1
LASH/2 LAST/T LATELY/T LATEST/5 LATRINE/2 LAURENT/1
LAVISH/2 LEA/2 LEARNED/2 LEARNING/T LECTURES/1 LEE/T
LEER/2 LEGERDEMAIN/2 LEK/2 LENAT/1 LEONARD/T LEPRECHAUN/2
LES/1 LESSER/1 LET/5 LET'S/5 LETTUCE/2 LEXICOMETRY/1
LEXICON/2 LIBIDINOUS/2 LIE/2 LIGHT/1 LIGHTFACE/2 LIKE/T
LIMBO/2 LIMIT/5 LIMITED/1 LINDA/1 LINE/1 LINEAR/1

LINEN/2 LINGUISTICS/T LINTEL/2 LISP/1 LIST/T LISTED/5
LISTEN/2 LISTING/5 LITHOGRAPHY/2 LIVELIHOOD/2 LOAN/2 LOCATE/2
LOCATION/1 LOCATIONS/1 LOGIC/1 LOGICAL/1 LONG/T LONGING/2
LORGNETTE/2 LOSING/1 LOVE/2 LOW/1 LUCID/2 LUMBER/2
LURID/2 LYRE/2 MACHINE/1 MACHINES/1 MACKINAW/2 MACRO/1
MADELINE/1 MAENAD/2 MAGAZINES/1 MAGNETIC/2 MAHOGANY/2 MAJORITY/2
MAKE/5 MALEFACTION/2 MALLET/2 MANAGEABLE/2 MANAGEMENT/1 MANGANESE/2
MANIFOLD/2 MANIPULATING/1 MANIPULATORS/T MANNA/1 MANSLAUGHTER/2 MANTRA/1
MANY/T MAPLE/2 MAPPING/1 MARCH/5 MARINATE/2 MARKET/1
MARQUETRY/2 MARR/1 MARSLAND/1 MARTELLI/1 MARTYR/2 MARVIN/T
MARY/1 MASINTER/1 MASSACHUSETTS/1 MASSEUR/2 MAT/2 MATCHING/T
MATRIARCHAL/2 MAUL/2 MAY/5 MCCARTHY/1 MCCORDUCK/1 MCDERMOTT/1
ME/T MEANING/1 MEANS/1 MEASURE/2 MEDIATION/2 MEDICAL/1
MEEK/2 MEETING/5 MEETINGS/5 MELODIC/2 MELTZER/1 MEMORIES/T
MEMORY/1 MENAGE/2 MENTION/T MENTIONED/T MENTIONING/5 MENTIONS/T
MENTOR/2 MENU/5 MENUS/T MERINGUE/2 MESSIAH/2 META-SYMBOLIC/1
METAMATHEMATICS METAPHORICAL/2 METHODS/1 METRE/2 MICHAEL/5 MICHALSKI/1
MICHIE/1 MICROMETER/2 MIDSUMMER/2 MIKE/1 MILIEU/2 MILLING/2
MINERAL/2 MINIMAL/1 MINKER/1 MINORITY/2 MINSKY/T MIRTH/2
MISCELLANY/2 MISFILE/2 MISRULE/2 MISTRIAL/2 MITCHELL/1 MLISP/1
MLISP2/1 MOBILE/2 MODEL/1 MODELING/1 MODELS/1 MODERNLY/2
MOIST/2 MOLLIFY/2 MONASTICISM/2 MONITOR/T MONKEY/1 MONOTONE/2
MONTH/5 MONTHS/5 MOOR/2 MORATORIUM/2 MORE/T MORPHEMIC/2
MORTIFICATION/2 MOST/5 MOSTOW/1 MOTILITY/2 MOTION/1 MOURN/2
MOVE/1 MOVEMENTS/1 MOVIES/1 MUCK/2 MULATTO/2 MULTILEVEL/1
MULTIPLE/2 MULTIPROCESS/1 MURDER/2 MUSIC/T MUSKETEER/2 MUST/T
MUTATIVE/2 MYNAH/2 MYSELF/5 N/2 NAGEL/1 NAP/2
NARRATOR/2 NASH-WEBBER/1 NATIONAL/1 NATIONALIZATION NATURAL/5 NAVAL/2
NEATH/2 NECROSIS/2 NEGOTIANT/2 NEOPLASM/2 NETS/5 NETTING/2
NETWORK/T NETWORKS/T NEURAL/T NEVER/2 NEW/T NEWBORN/1
NEWCOMER/1 NEWELL/T NEWEST/5 NEWEY/1 NEWSLETTER/5 NEXT/5
NICETY/2 NIH/1 NILS/T NILSSON/T NINE/1 NINETEEN/5
NINETY/5 NITRATE/2 NO/T NODDY/2 NOMINATING/1 NOMINATION/1
NOMINEES/1 NON-INDEPENDENT NONAGENARIAN/2 NONDETERMINISTIN NONPARTISAN/2 NOR/2
NORI/1 NORMAN/1 NOSTRIL/2 NOT/5 NOTES/1 NOTORIETY/2
NOVEMBER/5 NOWAY/2 NRL/1 NUGGET/2 NUN/2 O/2
OBEY/2 OBJECT/1 OBJECTS/1 OBLITERATE/2 OBSTACLE/2 OCCLUDE/2
OCTOBER/5 OCTOPUS/2 OF/T OFFENSIVE/2 OHLANDER/1 OHM/2
OK/1 OLDEST/T OMELETTE/2 ON/T ON-LINE/1 ONE/T
ONES/1 ONLY/T ONSLAUGHT/2 ONTOGENY/1 OPERA/2 OPERATIONAL/1
OPPOSITION/2 OPTIMAL/1 OPTIMIZED/1 OR/T ORANGUTAN/2 ORDER/1
ORDERS/1 ORDINAL/2 ORGANIZATION/1 ORGY/2 ORIENTED/1 ORNERY/2
OTTOMAN/2 OUR/5 OURSELVES/5 OUTCRY/2 OUTLANDISH/2 OUTRUN/2
OVATION/2 OVEREXPOSE/2 OVERLAYS/T OVERPASS/2 OVERTURN/2 OXIDIZE/2
PACK/2 PACKET/1 PAGODA/2 PAIR/1 PALATINE/2 PALLIATION/2
PAMELA/1 PAMPA/2 PANEL/2 PANTOMIME/2 PAPER/T PAPERS/T
PAPERT/1 PAPRIKA/2 PARALLEL/2 PARALLELISM/1 PARANOIA/1 PARAPHRASE/1
PARATYPHOID/2 PARIAH/2 PAROLE/2 PARRY/1 PARTHENOGENESIS PARTIAL/1
PAS/2 PASCAL/1 PASSKEY/2 PASTURE/2 PAT/1 PATHFINDER/1
PATIENCE/2 PATTERN/5 PAYMENT/2 PEARL/1 PECCARY/2 PEDICAB/2
PELAGE/2 PENDANT/2 PENNIES/2 PENULTIMATE/2 PERCENT/2 PERCEPTION/1
PERCEPTRONS/5 PERFECTION/2 PERFORMANCE/1 PERIPHRASIS/2 PERMISSIBLE/2 PERRY/1
PERSECUTE/2 PERSPECTIVE/2 PERVERT/2 PETER/1 PETITE/2 PHALANGES/2
PHASE/2 PHILOSOPHER/2 PHOSPHATE/2 PHOTOGRAMMETRY/PHOTOSYNTHESIS/PHRASE/T
PHRASES/5 PHYSICIANS/5 PIANIST/2 PICKLOCK/2 PICTURE/T PIECE/1
PIFFLE/2 PILL/2 PINCUSHION/2 PINGLE/1 PIONEER/2 PITUITARY/2
PLAGIARIZE/2 PLANES/1 PLANNER-LIKE/5 PLANNING/1 PLANS/1 PLANTATION/2
PLATYPUS/2 PLAYING/T PLEASE/T PLEBS/2 PLOD/2 PLUMY/2
PNEUMATIC/2 POINTED/2 POKER/1 POLICY/2 POLLOCK/2 POLYHEDRA/1
POLYP/2 PONCHO/2 POPE/2 PORCUPINE/2 PORTENTOUS/2 POSITIVE/2

POSTERIOR/2 POSTWAR/2 POTPOURRI/2 POX/2 PRASEODYMIUM/2 PRECARIOUS/2
PRECISION/2 PREDICAMENT/2 PREDICATE/5 PREFERABLY/2 PREFERENTIAL/T PREPONDERATE/2
PRESENT/T PRESENTIMENT/2 PRESUMPTION/2 PREVARICATE/2 PRICE/1 PRICE'S/1
PRIMARY/2 PRIMITIVES/1 PRINT/5 PRINTED/5 PRINTING/5 PRIOR/2
PRIVY/2 PROBLEM/T PROBLEMS/T PROCEDURAL/1 PROCEDURES/T PROCEEDING/5
PROCEEDINGS/5 PROCESSES/1 PROCESSING/1 PROCESSION/2 PRODUCE/5 PRODUCED/5
PRODUCTION/T PRODUCTIVITY/1 PROEM/2 PROGENY/2 PROGRAM/T PROGRAMMING/T
PROGRAMS/T PROGRESS/1 PROLIX/2 PROMULGATE/2 PROOF/T PROOFS/T
PROPERTIED/2 PROPERTIES/1 PROPULSION/2 PROSPEROUS/2 PROTESTATION/2 PROTOCOL/1
PROTOCOLS/1 PROVENIENCE/2 PROVER/1 PROVING/T PROXIMITY/2 PSYCHIATRIST/2
PSYCHOLOGY/5 PUBLISH/5 PUBLISHED/T PUBLISHER/5 PUBLISHERS/5 PUCK/2
PULE/2 PUMPERNICKEL/2 PUNT/2 PURIFIER/2 PURPOSE/1 PURSUE/2
PUTNAM/1 PUTRIDITY/2 QUADRUPLE/2 QUANTUM/2 QUEEN/2 QUERIES/T
QUESTION/1 QUIETUS/2 QUIT/5 QUIXOTIC/2 QUOTE/T QUOTED/T
RABBLE/2 RADIATE/2 RADIO/1 RAGE/2 RAJ/5 RAKISH/2
RALSTON/1 RAMAN/1 RANCIDITY/2 RAPHAEL/1 RAPIDLY/2 RASPBERRY/2
RATTAN/2 RAY/2 RAYMOND/1 REAL-WORLD/1 REALISM/2 REASONING/T
REBEL/2 RECEIPT/2 RECENT/T RECENTLY/5 RECITE/2 RECOGNITION/T
RECONNOITER/2 RECRUIT/2 REDDEN/2 REDDY/T REDUCTION/5 REED/1
REEVE/2 REFER/T REFERENCE/5 REFERENCED/5 REFERENCES/5 REFERRED/T
REFERRING/T REFRACTORY/2 REGARDING/T REGENCY/2 REGION/1 REGRESS/2
REGULARLY/1 REINCARNATION/2 REITER/1 RELATE/1 RELATED/T RELATES/1
RELATIONAL/1 RELEASED/5 RELUCTANCE/2 REMONSTRATE/2 RENEGADE/2 REPEAL/2
REPORT/5 REPORTER/1 REPORTERS/1 REPORTS/5 REPOSE/2 REPRESENTATION/
REPRESENTING/1 REPUDIATE/2 REQUEST/5 RESCIND/2 RESEARCH/1 RESIGNATION/2
RESOLUTION/T RESOURCE/1 RESPITE/2 RESPONSES/T RESTRICT/5 RESTRICTION/2
RETIRE/2 RETRIEVAL/5 RETRIEVE/5 REVEAL/2 REVIEWS/5 REVITALIZATION/
RHESUS/2 RHOMBERG/1 RIAL/2 RICH/1 RICHARD/1 RICK/1
RIDICULOUS/2 RIEGER/1 RIESBECK/1 RIM/2 RISEMAN/1 RISEN/2
RIVULET/2 ROBERT/1 ROBOT/T ROBOTIC/T ROBOTICS/1 ROBOTS/1
ROCHESTER/1 ROD/2 ROGER/1 ROMANTIC/2 RON/1 ROSEMARY/2
ROSENFELD/T ROTUNDA/2 ROUTE/2 RUBIN/1 RUBY/2 RULE/1
RULES/1 RUMANIAN/2 RUMELHART/1 RUSSET/2 RUTGERS/1 RYCHENER/1
S-L-GRAPHS/1 SABOTAGE/2 SACERDOTI/1 SADDLE/2 SAILOR/2 SALOON/2
SAMMET/1 SANCTITY/2 SANDEWALL/1 SANK/2 SAT/2 SATISFACTION/T
SATURDAY/2 SAVOR/2 SAY/1 SCANDINAVIAN/2 SCENE/1 SCENT/2
SCHANK/1 SCIENCE/T SCIENTIST/2 SCOTCH/2 SCOTT/1 SCREEN/2
SCUFFLE/2 SEAM/2 SEARCH/1 SECRET/2 SEDATIVE/2 SEE/T
SEEK/5 SEEKING/1 SEGMENTATION/1 SEGREGATION/2 SELECT/T SELTZER/1
SEMANTIC/T SEMANTICS/T SEMBLANCE/2 SEND/2 SENSE/T SENSUALLY/2
SENTENCE/1 SENTENCES/T SEPTEMBER/5 SEPTUAGINT/2 SERF/2 SERIAL/1
SESAME/2 SESSION/1 SESSIONS/5 SEVEN/1 SEVENTEEN/1 SEVENTY/5
SEVERAL/1 SEYMOUR/1 SHALL/2 SHAPE/1 SHAW/1 SHAWL/2
SHE/1 SHELLFISH/2 SHINGLE/2 SHOEMAKER/2 SHOOTING/1 SHORTLIFFE/1
SHOULD/T SHOW/5 SHOWER/2 SICKLE/2 SIGART/5 SIGNBOARD/2
SIKLOSSY/1 SILLY/2 SIMON/5 SIMPLIFY/2 SIMULATION/1 SIMULTANEOUS/1
SIMULTANEOUSLY/ SINCE/5 SIREN/2 SIX/1 SIXTEEN/5 SIXTY/5
SIZE/5 SKIFF/2 SLACK/2 SLAGLE/1 SLAVER/2 SLEW/2
SLITHER/2 SLOW/T SLUICE/2 SMC/1 SMILAX/2 SMITH/1
SMUG/2 SNARING/1 SNIDE/2 SNUFFLE/2 SO/5 SOBEL/1
SOCIALIZE/2 SOFTWARE/1 SOLARIUM/2 SOLIDLY/2 SOLOWAY/1 SOLUTIONS/5
SOLVING/T SOME/T SOMETHING/5 SOMEWHERE/1 SONATA/2 SORCERY/2
SORT/5 SORTS/T SOURCES/1 SPACE/5 SPADE/2 SPANNING/1
SPATE/2 SPECIOUS/2 SPEECH/5 SPEED/T SPENT/2 SPIRAL/2
SPLENDID/2 SPONTANEITY/2 SPRAWL/2 SPROULL/1 SPUNK/2 SRI/1
STAG/2 STALWART/2 STANFORD/1 STAPHYLOCOCCUS/ STATE/1 STATISTICIAN/2
STEAM/2 STEPBROTHER/2 STEREO/1 STEVE/1 STEVEDORE/2 STIMULUS/2
STOCHASTIC/1 STOCK/1 STOCKPILE/2 STOOL/2 STOP/5 STORAGE/1
STORED/5 STORIES/5 STORY/5 STRAIN/2 STREAM/2 STRIDING/2
STRONG/2 STRUCTURE/1 STRUCTURED/1 STRUCTURES/1 STUBBORN/2 STUDIES/1

STURGEON/2 SUBDUE/2 SUBJECT/6 SUBJECTS/6 SUBPROBLEMS/1 SUBSELECT/6
SUBSTANTIVE/2 SUBSYSTEM/1 SUBVERT/2 SUET/2 SUITABLE/2 SUMEX/1
SUMMARIES/5 SUMMARIZE/2 SUMMARY/5 SUNG/1 SUNSHINE/1 SUPERABUNDANT/2
SUPERVISE/2 SUPPOSED/2 SURE/1 SURGERY/2 SURNOTES/5 SURVEY/1
SURVEYING/2 SURVEYS/5 SUSSEX/1 SUZUKI/1 SYKES/1 SYLLABICATION/2
SYMBOL/1 SYMPHONY/2 SYNCHRONIZATIONSYNONYM/2 SYNTACTIC/1 SYNTAX/5
SYNTHESIS/T SYNTHESIZER/1 SYSTEM/T SYSTEMS/T TABLE/2 TACK/2
TAKE/T TALENT/2 TALES/1 TAMP/2 TANTALUM/2 TARGET/2
TASK/1 TATTING/2 TAXONOMY/2 TECH-II/1 TECHNICAL/1 TECHNICIAN/2
TECHNIQUES/1 TECHNOLOGY/1 TED/1 TELEGRAM/2 TELEOLOGICAL/1 TELL/5
TELLTALE/2 TEMPORAL/1 TEMPTATION/2 TEN/5 TENET/2 TERMINAL/1
TERMINALS/1 TERMINATE/5 TERMINATION/1 TERRIER/2 TERRY/T TESTIFY/2
TEXT/T TEXTURE/1 THANK/5 THANKS/1 THAT/T THATCH/2
THAUMATURGIST/1THE/T THEIR/5 THEM/T THEOREM/T THEORETICAL/2
THEORY/T THERE/T THERMOMETER/2 THESE/T THEY/5 THIEVES/2
THIRTEEN/5 THIRTY/1 THIS/T THOMAS/1 THORN/2 THORNDYKE/1
THOSE/T THOUGHT/1 THREE/1 THREW/2 THROUGH/5 THRUM/2
TIBETAN/2 TIGHT/2 TILL/1 TIME/5 TIMES/1 TIMPANI/2
TIPPET/2 TITLE/5 TITLED/2 TITLES/T TO/T TOENAIL/2
TOMAHAWK/2 TONIGHT/2 TOPER/2 TOPIC/5 TOPICS/T TOPOLOGY/1
TOROTH/2 TOTAL/2 TOUT/2 TRACE/2 TRAGEDY/2 TRANSACTION/5
TRANSACTIONS/5 TRANSATLANTIC/2TRANSFER/1 TRANSITION/T TRANSITIVE/2 TRANSMIT/5
TRANSMITTING/5 TRANSPORT/2 TREAT/2 TREES/1 TREPIDATION/2 TRICHINAE/2
TRILOGY/2 TRISECT/2 TROLL/2 TROUBLE/1 TROW/2 TRUNCHEON/2
TRY/5 TUBING/2 TUMOR/2 TURBOT/2 TURRET/2 TUTOR/1
TUTORIAL/1 TUTORING/1 TV/1 TWELVE/1 TWENTY/1 TWILIGHT/2
TWO/5 TYPES/1 TYPHOON/2 U.S./1 UHR/T ULLMAN/1
ULTRAMARINE/2 UNALLOYED/2 UNCERTAIN/2 UNCURL/2 UNDERESTIMATE/2UNDERPRODUCTION
UNDERSTANDING/5UNDERWATER/2 UNERRING/2 UNHARNESS/2 UNIFORM/T UNITARIAN/2
UNIVERSALS/1 UNMISTAKABLE/2 UNRAVEL/2 UP/T UPHEAVAL/2 UPROAR/2
URANIUM/2 URINALYSIS/2 US/5 USE/T USING/1 USSR/1
USUALLY/1 USURPATION/2 VACANT/2 VAIN/2 VALIDATION/2 VANADIUM/2
VAQUERO/2 VARIETY/1 VARSITY/2 VEERY/2 VELOCITY/2 VENEREAL/2
VERACIOUS/2 VERIFICATION/5 VERILY/2 VERSICLE/2 VERVE/2 VET/2
VIBRATION/2 VIC/1 VICTIMIZATION/2VIEWS/1 VIGIL/2 VILLUS/2
VIOLATION/2 VIRTUAL/2 VISCUS/2 VISION/T VISUAL/1 VIVIFIER/2
VOGUE/2 VOLUBLY/2 VOLUMES/5 VOTIVE/2 VULNERABILITY/2WAGER/2
WAITRESS/2 WALDINGER/1 WALLY/1 WAMPUM/2 WANT/5 WAREROOM/2
WARY/2 WAS/T WASN'T/1 WATERPOWER/2 WATSON/1 WAVEFORMS/T
WE/5 WE'D/5 WE'RE/1 WE'VE/1 WEAK/1 WEAKFISH/2
WEATHERWORN/2 WEIRD/2 WEIZENBAUM/1 WERE/5 WEREN'T/1 WHACK/2
WHAT/T WHAT'S/1 WHEN/T WHERE/T WHEREAS/2 WHICH/T
WHIFFLETREE/2 WHIPPOORWILL/2 WHITEFISH/2 WHO/T WHOLLY/2 WHY/5
WIDOW/2 WILFUL/2 WILKS/1 WILL/T WINDFALL/2 WINOGRAD/T
WINOGRAD'S/5 WINSTON/1 WINTERGREEN/2 WISH/5 WITH/1 WITHE/2
WORD/2 WOODS/T WOODY/1 WOOL/2 WORD/1 WORDS/1
WORK/5 WORLD/1 WORN/2 WOULD/1 WRANGLE/2 WRINKLE/2
WRITE/1 WRITING/T WRITTEN/T WROTE/T XENIA/2 YAMMER/2
YEAR/T YEARN/2 YEARS/5 YES/1 YON/2 YORICK/1
YOU/T YUMMY/2 ZINC/2 ZOHAR/1 ZOOID/2 ZUCKER/1

Appendix C: Schwa Deletion Rules

The following two tables summarize the schwa deletion rules used by Noah. Words from the 20,000 word dictionary were grouped according to the context about the schwa (stress of syllables, location of syllable boundaries and preceding and following phoneme). A rule is based on the very subjective test of whether the words of a group "sound right" for carefully articulated speech, when the schwa is deleted. This resulted in a schwa being deleted when it appears as 1) a one phoneme syllable (the first table) and 2) with one onset (the second table), for the left and right phoneme contexts given by the tables. (The right phoneme context appears at the top: L, M, N, and R; the left phoneme context appears at the left margin.) For both cases, the syllable preceding the syllable with the schwa must be stressed (indicated by a "1" before the syllable); the syllable following the syllable with the schwa must be unstressed (no number before the syllable).

One of three conditions are indicated for each position in the tables: 1) (no samples) -- No words were found with a schwa in the indicated context, 2) An underlined word with a pronunciation -- an example of a word for which the schwa is deleted by rule, and 3) A word which is not underlined, with a pronunciation -- an example of a word for the indicated context in which the schwa is not deleted. (Of course, there are many contexts not given by the table for schwas which were not deleted). Thus, rules are indicated in the table by <u>underlined words</u>.

Schwa's Appearing as -AX-

RIGHT: / LEFT:	L	M	N	R
B	PARABOLA P AX-1R AE B -AX-L AX	(NO SAMPLES)	CABINET 1K AE B -AX-N AX T	ROBBERY 1R AA B -AX-R IY
DH	(NO SAMPLES)	(NO SAMPLES)	(NO SAMPLES)	BRETHREN 1B R EH DH-AX-R AX N
D	PUDDLING 1P AX D -AX-L IH NX	ADAMANT 1AE D -AX-M AX N T	CARDINAL 1K AA R D -AX-N EL	FEDERAL 1F EH D -AX-R EL
F	SYPHILIS 1S IH F -AX-L AX S	(NO SAMPLES)	DEFINITE 1D EH F -AX-N AX T	REFERENCE 1R EH F -AX-R AX N S
G	NIGGLING 1N IH G -AX-L IH NX	BIGAMY 1B IH G -AX-M IY	AGONY 1AE G -AX-N IY	HAGGERY 1W AE G -AX-R IY
K	CHOCOLATE 1T SH AA K -AX-L AX T	MEDICAMENT ..1D IH K-AX-M AX N T	(NO SAMPLES)	HICKORY 1HH IH K -AX-R IY
M	FAMILY 1F AE M -AX-L IY	(NO SAMPLES)	(NO SAMPLES)	SUMMARY 1S AX M -AX-R IY
N	FINALLY 1F AY N -AX-L IY	MINIMAL 1M IH N -AX-M EL	(NO SAMPLES)	SCENERY 1S IY N -AX-R IY
P	HAPPLIY 1HH AE P -AX-L IY	(NO SAMPLES)	OPENING 1OW P -AX-N IH NX	SLIPPERY 1S L IH P -AX-R IY
R	MORALIST 1M OW R -AX-L AX S T	CARAMEL 1K AE R -AX-M EL	MARINER 1M AE R -AX-N ER	(NO SAMPLES)
SH	BACHELOR 1B AE T SH-AX-L ER	(NO SAMPLES)	NATIONAL 1N AE SH-AX-N EL	NATURAL 1N AE T SH-AX-R EL
S	DESOLATE 1D EH S -AX-L AX T	SPECIMEN 1S P EH S -AX-M AX N	LARCENY 1L AA R S -AX-N IY	CURSORY 1K ER S -AX-R IY
TH	CATHOLIC 1K AE TH-AX-L IH K	ANATHEMA AX-1N AE TH-AX-M AX	(NO SAMPLES)	PLETHORA 1P L EH TH-AX-R AX
V	JAVELIN 1D ZH AE V -AX-L AX N	(NO SAMPLES)	AVENUE 1AE V -AX-N Y UW	BRAVERY 1B R EY V -AX-R IY
ZH	MODULAR 1M AA D ZH-AX-L ER	REGIMENT 1R EH D ZH-AX-M AX N T	REGIONAL 1R IH D ZH-AX-N EL	SURGERY 1S ER D ZH-AX-R IY
Z	MEASLY 1M IY Z -AX-L IY	AZIMUTH 1AE Z -AX-M AX TH	PILSNER 1P IH L Z -AX-N ER	MISERY 1M IH Z -AX-R IY

Schwas Appearing as -<onset phoneme> AX -

When a schwa is deleted for the phoneme context given below, the left phoneme (i.e., the onset phoneme) is merged with the following or preceding syllable by the rule: if a legal onset (as defined by the onset lexicon given in Appendix B) is formed when the phoneme is appended to the onset of the following the syllable, the new onset is used in the pronunciation; otherwise, if a legal coda (as defined by the coda lexicon) is formed when the phoneme is appended to the coda of the preceding syllable, the new coda is used in the pronunciation; otherwise the schwa is not deleted. There is one exception to this: if the left phoneme is a "Y", it is deleted with the schwa. Again, the context, for which a schwa is deleted, is indicated by an underlined word.

RIGHT: / LEFT:	L	M	N	R
B	JUBILANT 1D ZH UW-B AX-L AX N T	(NO SAMPLES)	CONCUBINAGE ..1UW-B AX-N IH D ZH	NEIGHBORING 1N EY-B AX-R IH NX
D	INDOLENT 1IH N -D AX-L AX N T	ABDOMEN 1AE B -D AX-M AX N	MOLYBDENUM ..1L IH B -D AX-N EM	BOUNDARY 1B AW N -D AX-R IY
F	(NO SAMPLES)	INFAMOUS 1IH N -F AX-M AX S	SYMPHONY 1S IH M -F AX-N IY	OFFERING 1AO-F AX-R IH NX
G	PERGOLA 1P ER-G AX-L AX	(NO SAMPLES)	ORGANIST 1AO R -G AX-N AX S T	FINGERING 1F IH NX-G AX-R IH NX
K	VOCALIST 1V OW-K AX-L AX S T	ALCHEMIST 1AE L -K AX-M AX S T	BALCONY 1B AE L -K AX-N IY	BAKERY 1B EY-K AX-R IY
M	NORMALLY 1N AO R -M AX-L IY	ARMAMENT 1AA R -M AX-M AX N T	RUMINANT 1R UW-M AX-N AX N T	ADMIRABLE 1AE D-M AX-R AX-B EL
P	PURPLISH 1P ER-P AX-L IH SH	(NO SAMPLES)	TIMPANI 1T IH M -P AX-N IY	ASPIRIN 1AE S -P AX-R AX N
S	INSULIN 1IH N -S AX-L AX N	PROXIMATE 1P R AA K-S AX-M AX T	CONSONANCE 1K AA N-S AX-N AX N S	MENSURABLE 1M EH N-S AX-R AX-B EL
TH	(NO SAMPLES)	(NO SAMPLES)	(NO SAMPLES)	LUTHERAN 1L UW-TH AX-R AX N
T	(NO SAMPLES)	ESTIMABLE 1EH S-T AX-M AX-B EL	DESTINY 1D EH S -T AX-N IY	MYSTERY 1M IH S -T AX-R IY
Y	JOCULAR 1D ZH AA K-Y AX-L ER	DOCUMENT 1D AA K-Y AX-M AX N T	ALIENABLE 1EY L-Y AX-N AX-B EL	AUXILIARY AO G-1Z IH L-Y AX-R IY
Z	CAUSALLY 1K AO-Z AX-L IY	(NO SAMPLES)	(NO SAMPLES)	ROSARY 1R OW-Z AX-R IY

Appendix E: Training and Test Utterances

Training Utterances -- 174 Utterances

Training Set LAA -- 20 utterances

PLEASE HELP ME
WHAT SHOULD I ASK
WHAT CAN THE SYSTEM DO
THE FIRST TWO
GIVE ME ONE MORE PLEASE
THANK YOU I'M DONE
STOP TRANSMITTING PLEASE
WHO WROTE IT
WHO WAS THE AUTHOR
WHAT WAS ITS TITLE
WHEN WAS IT PUBLISHED
WHAT ABOUT MINSKY
WHICH IS THE OLDEST
WHAT FACTS ARE STORED
PLEASE LIST THE AUTHORS
PRINT THE NEXT ONE
WHERE DOES HE WORK
WHAT IS HER AFFILIATION
WHAT ABOUT FORMAL SEMANTICS
WHAT ABOUT PROGRAM VERIFICATION

Training Set LAB -- 20 utterances

ARE ANY ARTICLES BY REDDY
WHAT HAS DREYFUS WRITTEN LATELY
LIST THE ABSTRACTS BY NEWELL OR SIMON
DO ANY PAPERS CITE NILSSON
DO MANY ABSTRACTS DISCUSS SYNTAX
HOW MANY PAPERS REFER TO FRAME THEORY
WHERE IS PREDICATE CALCULUS MENTIONED
ARE NEURAL NETWORKS MENTIONED ANYWHERE
DO ANY OF THESE MENTION PSYCHOLOGY
IS HEURISTIC PROGRAMMING MENTIONED
WHO HAS WRITTEN ABOUT PATTERN MATCHING
WHEN WAS THAT BOOK WRITTEN
GIVE ME THE DATE OF THAT ABSTRACT
WHAT IS THE TITLE OF THAT PAPER
WHAT IS THE SIZE OF THE DATA BANK
WHAT ADDRESS IS GIVEN FOR THE AUTHORS
GIVE THE AUTHOR AND DATE OF EACH
HOW MANY REFERENCES ARE GIVEN
PLEASE MAKE ME A FILE OF THOSE
CAN I HAVE THESE ABSTRACTS LISTED

Training Set LAC -- 20 utterances

DID ANY AI JOURNAL PAPERS CITE WOODS
ARE ANY BY UHR
THE AREA I'M INTERESTED IN IS UNDERSTANDING
WHAT ARE SOME OF THE AREAS OF ARTIFICIAL INTELLIGENCE
ARE YOU ALWAYS THIS SLOW
WHAT CAN I DO TO SPEED YOU UP
AREN'T THERE ANY ABSTRACTS SINCE NINETEEN SEVENTY FIVE
LET'S RESTRICT OUR ATTENTION TO PAPERS SINCE NINETEEN SEVENTY FOUR
WHAT SORTS OF RECOGNITION DEVICES ARE WRITTEN UP
DO ANY OF THESE ALSO MENTION PATTERN RECOGNITION
DOES PATTERN DIRECTED FUNCTION INVOCATION GET MENTIONED ANYWHERE
IS RESOLUTION THEOREM PROVING MENTIONED IN AN ABSTRACT
HOW MANY OF THESE ALSO DISCUSS ABSTRACTION
ANY ABSTRACTS REFERRING TO DYNAMIC CLUSTERING
WHEN WAS CELL ASSEMBLY THEORY LAST REFERRED TO
WHICH COGNITIVE PSYCHOLOGY CONTAINS WINOGRAD'S ARTICLE
DON'T GET ME ANY ARTICLES WHICH MENTION GAME PLAYING
DOES THAT ARTICLE MENTION TIME OR SPACE BOUNDS
WHICH PAPERS ON LANGUAGE UNDERSTANDING ARE ABOUT ENGLISH
WHICH PAPERS ON CONTROL ALSO DISCUSS GRAIN OF COMPUTATION

Training Set LAD -- 34 utterances

DO ANY PAPERS CITE NILSSON
HAVE ANY NEW PAPERS BY NEWELL APPEARED
DO YOU HAVE ANY NEW PAPERS ON SPEECH UNDERSTANDING
GIVE ME THE DATE OF THAT ABSTRACT
HOW MANY PAPERS REFER TO FRAME THEORY
I AM INTERESTED IN LANGUAGE UNDERSTANDING
DO MANY ABSTRACTS DISCUSS SYNTAX
WHAT IS THE TITLE OF THAT PAPER
WHO WROTE IT
IS HEURISTIC PROGRAMMING MENTIONED
LIST THE ABSTRACTS BY NEWELL OR SIMON
GIVE THE AUTHOR AND DATE OF EACH
THE FIRST TWO
PRINT THE NEXT ONE
HOW MANY PAPERS DISCUSS HILL CLIMBING
DO ANY OF THESE ALSO MENTION PATTERN RECOGNITION
ARE ANY BY UHR
WHAT ABOUT MINSKY
WHAT IS THE TITLE OF THE MOST RECENT ONE
WHICH ARTICLES REFER TO THESE
ARE ANY ARTICLES BY REDDY
WHICH IS THE OLDEST
HOW MANY ABSTRACTS ARE THERE ON PROBLEM SOLVING
DO ANY PAPERS DISCUSS PLANNER-LIKE LANGUAGES
ARE THERE ANY RECENT ARTICLES IN CACM
WHERE DID THAT ARTICLE APPEAR
WHEN WAS IT PUBLISHED
WHAT HAS DREYFUS WRITTEN LATELY
WHO HAS WRITTEN ABOUT PATTERN MATCHING
IS THERE ANYTHING NEW REGARDING SEMANTIC NETS
WHAT ABOUT FORMAL SEMANTICS
HAVE ANY ARTICLES APPEARED WHICH MENTION HEARSAY
WHAT ARE THE TITLES OF THE RECENT ARPA SURNOTES
HOW MANY ARTICLES ON PRODUCTION SYSTEMS ARE THERE

Training Set LMA -- 20 utterances

WHICH SUMMARIES ON AI CONSIDER PATTERN RECOGNITION IN ADDITION
WHAT ARE THEIR AFFILIATIONS
WHAT ADDRESSES ARE GIVEN FOR THE AUTHORS
WHAT ISSUES DURING JANUARY AND JULY CONCERN CONTROL
LET US CONFINE OURSELVES TO JOURNALS AFTER FEBRUARY NINETEEN FIFTY
TELL ME THE TITLES OF THE EARLIEST TEN
CHOOSE AMONG VOLUMES BEFORE NINETEEN SIXTY
WHICH OF THESE APPEARED RECENTLY IN THE IEEE TRANSACTIONS
HOW MANY BOOKS WERE PRODUCED FROM MARCH TO DECEMBER
HOW BIG IS THE DATA BASE
QUIT LISTING PLEASE
CEASE PRINTING
LIST THE NEXT FOURTEEN HUNDRED
IS THERE AN IFIP CONVENTION ISSUE FROM MAY OR JUNE
I DEMAND ANOTHER ARTICLE AFTER AUGUST NINETEEN THIRTEEN
DID THE SIGART NEWSLETTER PUBLISH ANYTHING IN OCTOBER OR NOVEMBER
WE WANT SOME REVIEWS CONCERNING PERCEPTRONS
DID NEWELL PRESENT A PAPER AT THE IFIP MEETINGS IN SEPTEMBER
HOW MANY PAPERS FROM APRIL THROUGH AUGUST CONCERNED CHESS
WE'D LIKE TO SEE THE TITLES FROM PROCEEDINGS OF THE ACM CONFERENCE

Training Set LMB -- 20 utterances

GENERATE A COPY OF THOSE
WE DESIRE A PROCEEDING OF THE ACM MEETING REFERENCED BY NEWELL
COULD YOU RETRIEVE SOMETHING FROM INFORMATION AND CONTROL DISCUSSING AI
DID ANYONE PUBLISH ABOUT LEARNING IN COMMUNICATIONS OF THE ACM
FINISH PRINTING
WHAT ARE THE KEY PHRASES
I'D LIKE TO SEE THE MENUS
HAVEN'T YOU FINISHED
WHICH STORIES IN THE SIGART NEWSLETTER HAVE BEEN DISCUSSING CONTROL
TRANSMIT THE NEXT EIGHTEEN
HASN'T LANGUAGE UNDERSTANDING BEEN CONSIDERED IN COMPUTING REVIEWS
LET ME LIMIT MYSELF TO REPORTS ISSUED SINCE NINETEEN FIFTEEN
HASN'T A CURRENT REPORT ON SPEECH UNDERSTANDING BEEN RELEASED
DIDN'T THAT PAPER QUOTE NILSSON
ARE NOT SOME OF THESE FROM COMPUTING SURVEYS
DOESN'T THIS PAPER REFERENCE AN IEEE TRANSACTION
WHY IS THE SYSTEM SO SLOW
KILL THE LISTING
WHAT KINDS OF SUBJECTS ARE STORED
WHICH SORT OF RETRIEVAL KEYS CAN I SEEK

Training Set LMC -- 20 utterances

SELECT FROM ARTICLES ON LANGUAGE UNDERSTANDING
SUBSELECT FROM GAME PLAYING
WHAT SUBJECT CAN I REQUEST
WE WISH TO GET THE LATEST FORTY ARTICLES ON ASSOCIATIVE MEMORIES
GIVE ME SOMETHING MENTIONING ABSTRACTION
PLEASE TERMINATE TRANSMITTING
WHAT SORT OF SUMMARY IS AVAILABLE
I'D LIKE TO KNOW THE PUBLISHERS OF THAT STORY
SHOW ME ITS PUBLISHER
WHAT TOPIC MENU CAN I CHOOSE
SHOW ME THE LATEST ELEVEN

```
ARE ANY OF THESE FROM THE IFIP SESSIONS IN THE MONTH OF JUNE
THE LATEST SIXTEEN PLEASE
DURING WHAT MONTHS WERE THEY PUBLISHED
WHO WAS QUOTED IN THAT ARTICLE
PRODUCE A COPY OF THE NEWEST EIGHTY ARTICLES
DO ANY RECENT ACM CONFERENCES CONSIDER PSYCHOLOGY
DID ANY IEEE CONVENTIONS PUBLISH PROCEEDINGS
WAS NEWELL CITED BY ANY REPORTS ISSUED IN THE LAST NINETY YEARS
TRY TO GET SURVEYS PRINTED IN THE LAST EIGHTY MONTHS
```

Training Set LLA -- 20 utterances

```
DO ANY PAPERS CITE MICHAEL ARBIB
HAVE ANY NEW PAPERS BY ISSAC ASIMOV APPEARED
DO YOU HAVE NEW PAPERS ON ACQUISITION OF KNOWLEDGE
HOW MANY PAPERS REFER TO ACTIVE KNOWLEDGE
I AM INTERESTED IN ADAPTATION
DO MANY ABSTRACTS DISCUSS AN ADAPTIVE NATURAL LANGUAGE SYSTEM
IS ALGEBRAIC REDUCTION MENTIONED
LIST THE ABSTRACTS BY RAJ REDDY OR HARRY BARROW
HOW MANY PAPERS DISCUSS ALGOL
DO ANY OF THESE ALSO MENTION AUTOMATIC CODING
ARE ANY BY HANS BERLINER
WHAT ABOUT DANNY BOBROW
ARE ANY ARTICLES BY BRUCE BUCHANAN
HOW MANY ABSTRACTS ARE THERE ON ADAPTIVE PRODUCTION SYSTEMS
DO ANY PAPERS DISCUSS ADVISING PHYSICIANS
WHAT HAS HERB SIMON WRITTEN LATELY
WHO HAS WRITTEN ABOUT ALGORITHMIC AESTHETICS
IS THERE ANYTHING NEW REGARDING ALL-OR-NONE SOLUTIONS
WHAT ABOUT ANALOGY IN PROBLEM SOLVING
HOW MANY ARTICLES ON ANALYSIS OF CONTEXT ARE THERE
```

Testing Utterances -- 105 Utterances

Noah's performance for each utterance is given for the 1000-word vocabulary. A dashed line indicates that the word was not hypothesized; a number after a word gives the rank of the hypothesis.

Test Set LAF - 25 utterances

```
1. (WHAT   PAPERS   ON   GRAMMATICAL   INFERENCE   ARE   THERE   )
   WHAT 4 PAPERS 1 --   GRAMMATICAL 1 INFERENCE18 ARE11 THERE 2
2. (ANY   ABSTRACTS   REFERRING   TO   DYNAMIC   CLUSTERING   )
   ANY 1 ABSTRACTS 1 REFERRING 1 TO 3 DYNAMIC 1 ----------
3. (WHICH   PAPERS   CITE   FEIGENBAUM   AND   FELDMAN   )
   WHICH 1 PAPERS 1 CITE 2 ----------   AND 7 -------
4. (ARE   THERE   ANY   NEW   PAPERS   ON   GRAPH   MATCHING   )
   ARE12 THERE 2 ANY 1 NEW 8 PAPERS 1 ON 3 -----   MATCHING 5
5. (IS   RESOLUTION   THEOREM   PROVING   MENTIONED   IN   AN   ABSTRACT   )
   IS 1 ----------   -------   -------   MENTIONED 2 IN 2 AN 5 ABSTRACT 5
6. (GET   ME   EVERYTHING   ON   UNIFORM   PROOF   PROCEDURES   )
   ---   ME 1 ----------   --   UNIFORM 7 PROOF 2 PROCEDURES 1
7. (NO   MORE   PLEASE   )
   NO 1 MORE15 PLEASE 2
8. (GIVE   ME   ONE   MORE   PLEASE   )
   GIVE18 ME 1 ONE 3 MORE 1 PLEASE 1
9. (WHO   WROTE   PAPERS   ON   PRODUCTION   SYSTEMS   THIS   YEAR   )
```

```
    WHO 2 WROTE 1 PAPERS 1 ON 1 PRODUCTION 1 -------  THIS 3 ----
18. (DID   ANY   AI   JOURNAL   PAPERS   CITE   WOODS   )
     ---   ANY 1 AI 1 JOURNAL 1 PAPERS 3 CITE 1 -----
11. (DO   ALL   QUERIES   TAKE   THIS   LONG   )
     --   ALL 1 -------   TAKE 3 THIS10 LONG 9
12. (ARE   YOU   ALWAYS   THIS   SLOW   )
     ARE 1 ---   ------   THIS 2 SLOW 3
13. (HOW   LONG   DOES   IT   TAKE   )
     ---   ----   DOES 1 IT 1 TAKE 2
14. (WHEN   WILL   YOU   HAVE   THE   ANSWER   )
     WHEN 3 WILL 3 ---   HAVE 8 ---   ANSWER 5
15. (DOES   IT   ALWAYS   TAKE   THIS   LONG   TO   ANSWER   ME   )
     ----   IT 2 ALWAYS 3 ----   THIS 4 ----   TO 4 ANSWER 1 ME 2
16. (DO   RESPONSES   EVER   COME   FASTER   )
     DO 4 RESPONSES 7 ----   COME 1 FASTER12
17. (WHAT   CAN   I   DO   TO   SPEED   YOU   UP   )
     WHAT 2 CAN 1 I 3 DO 1 TO 1 SPEED 1 YOU 6 UP 4
18. (HOW   CAN   I   USE   THE   SYSTEM   EFFICIENTLY   )
     ---   CAN 2 I 6 USE 3 THE 2 SYSTEM 2 ----------
19. (WHAT   MUST   I   ASK   )
     ----   ----   I 1 ASK 2
20. (WHAT   DO   I   HAVE   TO   DO   )
     WHAT 1 --   I 4 HAVE 1 TO 1 DO 1
21. (HELP   )
     HELP 3
22. (CAN   YOU   HELP   ME   )
     CAN 1 YOU 2 ----   ME 1
23. (PLEASE   HELP   ME   )
     PLEASE 1 HELP 3 ME 1
24. (WHAT   SHOULD   I   ASK   )
     WHAT 5 SHOULD 3 I 2 ---
25. (WHEN   WAS   THE   LAST   PAPER   BY   HOLLAND   PUBLISHED   )
     WHEN 7 WAS 3 THE 1 LAST 1 PAPER 1 BY 3 -------   ---------
```

Test Set LMN -- 20 utterances

```
1. (ANY   ABSTRACTS   REFERRING   TO   AI   OR   ARTIFICIAL   INTELLIGENCE   )
    ANY 2 ABSTRACTS 7 REFERRING 1 TO 9 AI 1 OR 3 ----------  ------------
2. (ARE   ASSOCIATIVE   MEMORIES   DISCUSSED   IN   RECENT   JOURNALS   )
    ARE 2 -----------   --------   DISCUSSED 3 IN 4 RECENT 3 JOURNALS 1
3. (ARE   LEARNING   AND   NEURAL   NETWORKS   MENTIONED   ANYWHERE   )
    ARE 2 LEARNING 5 AND 8 ------   NETWORKS 1 MENTIONED 2 ANYWHERE 2
4. (DID   REDDY   PRESENT   A   PAPER   AT   IJCAI   )
    DID 6 REDDY 7 -------   A 1 PAPER 1 --   -----
5. (DION'T   THAT   PAPER   QUOTE   DREYFUS   )
    ------   THAT 2 PAPER 1 -----   DREYFUS 1
6. (DOES   PICTURE   RECOGNITION   GET   MENTIONED   ANYWHERE   )
    DOES 5 PICTURE 1 RECOGNITION 1 GET 2 MENTIONED 2 ANYWHERE 2
7. (GET   ME   EVERYTHING   ON   DYNAMIC   CLUSTERING   )
    ---   ME 1 ----------   ON 6 DYNAMIC 1 ----------
8. (GENERATE   A   COPY   OF   THOSE   )
    GENERATE 1 -   COPY 5 OF 2 THOSE13
9. (GIVE   ME   THE   DATE   OF   THAT   ABSTRACT   )
    GIVE19 ME 3 THE 3 DATE 5 OF 2 THAT 1 ABSTRACT 1
10. (HOW   CAN   I   USE   THE   SYSTEM   EFFICIENTLY   )
     HOW 1 CAN 1 I 9 USE 3 THE 1 SYSTEM 5 EFFICIENTLY 2
11. (I   AM   INTERESTED   IN   LEARNING   )
     I 6 --   INTERESTED 7 IN 3 LEARNING 3
12. (I'D   LIKE   TO   SEE   THE   MENUS   )
     I'D 1 LIKE 1 TO 1 SEE 1 THE 1 MENUS 1
```

```
13. (SELECT   FROM   ARTICLES   ON   GAME   PLAYING   )
    ------   FROM 1 ARTICLES 1 ON13 ----   PLAYING 1
14. (WHAT   ADDRESSES   ARE   GIVEN   FOR   THE   AUTHORS   )
    WHAT 1 ---------   ---   GIVEN 1 FOR 3 THE 2 -------
15. (WHAT   PAPERS   ON   PREFERENTIAL   SEMANTICS   ARE   THERE   )
    WHAT 1 PAPERS 1 ON 4 ------------   SEMANTICS 4 ARE 8 THERE 3
16. (WHEN   WAS   A   SEMANTIC   NETWORK   LAST   REFERRED   TO   )
    WHEN 1 WAS10 -   --------   NETWORK 3 LAST10 REFERRED 1 TO 1
17. (WHICH   PAPERS   CITE   FELDMAN   )
    WHICH 1 PAPERS 1 CITE 3 -------
18. (WHO   HAS   WRITTEN   ABOUT   AUTOMATIC   PROGRAMMING   )
    WHO 4 ---   -------   ABOUT 1 ---------   PROGRAMMING 6
19. (WHO   WAS   QUOTED   IN   THAT   ARTICLE   )
    WHO 1 WAS 2 ------   IN 4 THAT14 -------
20. (WHICH   IS   THE   OLDEST   )
    -----   IS 3 THE 5 OLDEST 5

Test Set LLB -- 20 utterances
1. (DO   ANY   OF   THESE   MENTION   ANALYSIS   OF   SENTENCES   )
   --   ANY 1 OF 2 THESE 1 MENTION 1 --------   OF 1 ---------
2. (WHICH   AI   TEXT   CONTAINED   THE   ARTICLE   BY   ALLEN   NEWELL   )
   WHICH 2 AI 1 ----   CONTAINED 1 THE 6 ARTICLE 1 BY 2 -----   NEWELL 1
3. (WHAT   TOPICS   ARE   RELATED   TO   AUTOMATIC   PROGRAMMING   )
   WHAT 2 ------   ARE 7 -------   TO 3 ---------   -----------
4. (DOES   ASSIMILATION   OF   NEW   INFORMATION   GET   DISCUSSED   ANYWHERE   )
   DOES 3 ASSIMILATION 1 OF 3 NEW 1 INFORMATION 1 GET16 DISCUSSED 2 --------
5. (WHICH   TITLES   CONTAIN   THE   PHRASE   AXIOMS   FOR   GO   )
   -----   TITLES 1 CONTAIN 1 THE 1 PHRASE 1 ------   FOR 1 --
6. (DOES   THAT   ARTICLE   MENTION   AXIOMATIC   SEMANTICS   )
   DOES 1 THAT 6 ARTICLE 1 MENTION 1 ---------   SEMANTICS 1
7. (WHICH   OF   THEM   DISCUSSES   AUTOMATED   DEDUCTION   )
   WHICH 1 OF 4 ----   DISCUSSES 1 AUTOMATED 1 DEDUCTION 1
8. (ARE   THERE   ANY   ABSTRACTS   WHICH   REFER   TO   PAPERS   BY   BILL   WOODS   )
   ---   THERE 4 ANY 1 ABSTRACTS 1 -----   REFER 1 TO 7 PAPERS 1 BY 4 ----   -----
9. (WHERE   IS   AUTOMATIC   COMPUTATION   AND   CONTROL   MENTIONED   )
   WHERE 4 IS 3 AUTOMATIC 1 -----------   AND 2 -------   MENTIONED 2
10. (WHAT   ARE   SOME   OF   THE   AREAS   OF   COGNITIVE   SCIENCE   )
    WHAT 2 ARE 4 SOME 1 OF 7 THE10 AREAS 6 OF 1 ---------   -------
11. (ARE   ANY   ARTICLES   ABOUT   BIOMEDICINE   )
    ARE 2 ANY 2 ARTICLES 3 ABOUT 3 -----------
12. (DO   ANY   OF   THE   ABSTRACTS   MENTION   AUGMENTED   TRANSITION   NETWORKS   )
    DO 3 ANY 2 OF 4 THE 3 ---------   MENTION 1 AUGMENTED 4 TRANSITION 1 --------
13. (HOW   MANY   OF   THESE   ALSO   DISCUSS   AUTOMATIC   PROGRAM   WRITING   )
    HOW 1 MANY 1 OF 3 -----   ALSO 1 DISCUSS 1 ---------   PROGRAM 9 WRITING 2
14. (WHICH   PAPERS   ON   BELIEF   SYSTEMS   ARE   ABOUT   CAUSAL   REASONING   )
    WHICH 1 PAPERS 1 ON 7 ------   SYSTEMS 1 ARE 4 -----   ------   REASONING 2
15. (DO   ANY   PAPERS   ON   AUTOMATIC   PROOF   OF   CORRECTNESS   EXIST   )
    DO 2 ANY 1 PAPERS 1 ON 4 AUTOMATIC 1 PROOF 1 OF 9 -----------   -----
16. (WHAT   ABOUT   AUTOMATIC   PROGRAM   SYNTHESIS   FROM   EXAMPLE   PROBLEMS   )
    WHAT 1 ABOUT 1 ---------   -------   ---------   FROM 2 -------   --------
17. (I   AM   INTERESTED   IN   COGNITION   )
    I 2 --   INTERESTED 1 IN 3 ---------
18. (THE   AREA   I   AM   INTERESTED   IN   IS   AUTOMATION   )
    THE10 AREA 1 I 2 --   INTERESTED 1 IN 3 IS12 AUTOMATION 1
19. (DON'T   GET   ME   ANY   ARTICLES   WHICH   MENTION   BACKGAMMON   )
    -----   GET 1 ME 7 ANY 2 ARTICLES 1 WHICH 2 MENTION 2 BACKGAMMON 1
20. (I   AM   ONLY   INTERESTED   IN   PAPERS   ON   BINDINGS   )
    I 2 --   ONLY 9 INTERESTED 1 IN 1 PAPERS 1 ON 4 --------
```

```
Test Set LLC -- 20 utterances
1. (DO   ANY   PAPERS   THIS   YEAR   CITE   JOHN   HOLLAND   )
   --   ANY 1 PAPERS 1 THIS 3 YEAR 8 CITE 1 JOHN 4 -------
2. (WHAT   PAPERS   ON   AUTOMATIC   THEOREM   PROVING   ARE   THERE   )
   WHAT 2 PAPERS 1 ON 8 AUTOMATIC 1 -------   -------   ARE 4 -----
3. (ANY   ABSTRACTS   REFERRING   TO   THE   BERKELEY   DEBATE   )
   ANY 1 ABSTRACTS 5 ---------   TO 1 THE 1 --------.   ------
4. (WHICH   PAPERS   CITE   AZRIEL   ROSENFELD   )
   WHICH 1 PAPERS 1 CITE 2 ------   ---------
5. (ARE   THERE   ANY   NEW   PAPERS   ON   BUSINESS   PROBLEM   SOLVING   )
   ARE 1 THERE 1 ANY 1 ---   PAPERS 1 ON 6 BUSINESS 1 PROBLEM 1 -------
6. (IS   THE   BAY   AREA   CIRCLE   MENTIONED   IN   AN   ABSTRACT   )
   IS 1 THE 1 BAY 8 ----   ------   ---------   IN 1 AN 2 ABSTRACT 1
7. (GET   ME   EVERYTHING   ON   CARTOGRAPHY   )
   GET 6 ME13 ----------   ON 8 ----------
8. (WHO   WROTE   PAPERS   ON   BRAIN   THEORY   THIS   YEAR   )
   WHO 1 WROTE 1 PAPERS 1 ON 3 -----   ------   THIS 3 YEAR 4
9. (DID   ANY   ACL   PAPERS   CITE   TERRY   WINOGRAD   )
   DID 2 ANY 1 ---   PAPERS 1 CITE 2 -----   WINOGRAD 7
10. (WHEN   WAS   THE   LAST   PAPER   BY   MARVIN   MINSKY   PUBLISHED   )
    WHEN 2 WAS 1 ---   LAST 4 PAPER 1 BY 5 ------   ------   PUBLISHED 1
11. (WHEN   WAS   CIRCUIT   ANALYSIS   LAST   REFERRED   TO   )
    WHEN 1 WAS 4 CIRCUIT 1 --------   LAST 1 REFERRED 1 TO 2
12. (ARE   CASE   SYSTEMS   MENTIONED   ANYWHERE   )
    ARE 3 CASE15 SYSTEMS 1 MENTIONED 2 ANYWHERE 2
13. (DO   ANY   OF   THOSE   PAPERS   MENTION   CHECKING   PROOFS   )
    DO 2 ANY 1 OF 2 THOSE 8 PAPERS 1 MENTION 2 CHECKING 1 ------
14. (WHICH   PAPERS   ON   CHESS   PLAYING   PROGRAMS   ALSO   DISCUSS   COMMON   SENSE   )
    WHICH 1 PAPERS 1 ON 8 CHESS 6 -------   --------   ALSO 1 DISCUSS 1 COMMON 1 SENSE12
15. (WHAT   SORTS   OF   COGNITIVE   ROBOTIC   SYSTEMS   ARE   WRITTEN   UP   )
    WHAT 1 -----   OF 7 ---------   ROBOTIC 2 SYSTEMS 1 ARE 1 -------   --
16. (DO   ANY   AUTHORS   DESCRIBE   COMMON   SENSE   THEORY   FORMATION   )
    DO 1 ---   -------   --------   COMMON 1 SENSE10 ------   FORMATION 1
17. (WAS   IT   PUBLISHED   BY   THE   ASSOCIATION   FOR   COMPUTATIONAL   LINGUISTICS   )
    WAS 2 IT 2 ---------   BY 2 ---   ASSOCIATION 2 FOR 1 ------------   LINGUISTICS 1
18. (IS   THAT   ABOUT   COMPLEX   WAVEFORMS   )
    IS 3 THAT 1 ABOUT 1 COMPLEX 1 WAVEFORMS13
19. (WHICH   PAPER   MENTIONS   AN   ASSEMBLY   ROBOT   )
    WHICH 1 PAPER 1 MENTIONS 1 AN 1 ASSEMBLY 9 ROBOT 2
20. (IS   AN   AXIOMATIC   SYSTEM   REFERRED   TO   )
    --   AN 2 ---------   SYSTEM 1 REFERRED 1 TO 9

Test Set LLD -- 20 utterances
1. (DO   ANY   PAPERS   CITE   ED   FEIGENBAUM   )
   DO 1 ANY 1 PAPERS 1 CITE 1 ED10 FEIGENBAUM 6
2. (HAVE   ANY   NEW   PAPERS   BY   JERRY   FELDMAN   APPEARED   )
   HAVE 1 ANY 1 NEW 1 PAPERS 1 BY 3 -----   -------   APPEARED 1
3. (DO   YOU   HAVE   NEW   PAPERS   ON   A   CAI   MONITOR   )
   DO 8 ---   HAVE 2 NEW 1 PAPERS 1 ON 9 A 2 CAI 1 MONITOR 1
4. (HOW   MANY   PAPERS   REFER   TO   A   COMMON   SENSE   ALGORITHM   )
   HOW 1 MANY 1 PAPERS 2 REFER 1 --   A 5 COMMON 1 SENSE14 ---------
5. (I   AM   INTERESTED   IN   COMPUTATIONAL   LINGUISTICS   )
   I 2 --   INTERESTED 1 IN 2 -------------   LINGUISTICS13
6. (DO   MANY   ABSTRACTS   DISCUSS   COMPUTER   ART   )
   DO 4 MANY 3 ABSTRACTS 1 -------   --------   ART12
7. (IS   COMPUTER   MUSIC   MENTIONED   )
   IS 1 COMPUTER10 -----   MENTIONED 3
8. (LIST   THE   ABSTRACTS   BY   LEONARD   UHR   )
   LIST 4 THE 2 ABSTRACTS 1 BY 1 -------   ---
```

```
9. (HOW   MANY   PAPERS   DISCUSS   COMPUTER   CONTROLLED   MANIPULATORS   )
   HOW 4 MANY 1 PAPERS 2 DISCUSS 2 COMPUTER 2 CONTROLLED 6 ------------
10. (DO   ANY   OF   THESE   ALSO   MENTION   COMPUTER   GRAPHICS   )
    --   ---   OF 2 THESE 1 ALSO 1 MENTION 1 --------   GRAPHICS 1
11. (ARE   ANY   BY   NILS   NILSSON   )
    ARE11 ANY 1 BY 1 ----   NILSSON 7
12. (WHAT   ABOUT   KEN   COLBY   )
    WHAT 1 ABOUT 1 KEN 1 COLBY 8
13. (ARE   ANY   ARTICLES   BY   ALLEN   COLLINS   )
    ARE 3 ANY 3 ARTICLES 4 BY 1 -----   -------
14. (HOW   MANY   ABSTRACTS   ARE   THERE   ON   COMPUTER   VISION   )
    HOW 1 MANY 3 ABSTRACTS 3 ARE 1 -----   ON14 --------   ------
15. (DO   ANY   PAPERS   DISCUSS   COMPUTER   BASED   CONSULTATIONS   )
    DO 2 ANY 1 PAPERS 2 DISCUSS 1 COMPUTER 1 -----   CONSULTATIONS 2
16. (WHAT   HAS   LEE   ERMAN   WRITTEN   LATELY   )
    WHAT 1 HAS 1 LEE 5 ERMAN 1 -------   ------
17. (WHO   HAS   WRITTEN   ABOUT   CONCEPTUAL   DESCRIPTIONS   )
    ---   ---   --------   ABOUT 1 ----------   ------------
18. (IS   THERE   ANYTHING   NEW   REGARDING   CONCEPTUAL   INFERENCE   )
    IS 1 THERE 1 --------   NEW 1 REGARDING 6 ----------   INFERENCE 3
19. (WHAT   ABOUT   CONCEPTUAL   OVERLAYS   )
    WHAT 1 ABOUT 2 ----------   --------
20. (HAVE   ANY   ARTICLES   APPEARED   WHICH   MENTION   CONSTRAINT   SATISFACTION   )
    HAVE 1 ANY 2 ARTICLES 2 --------   WHICH 1 -------   ----------   SATISFACTION 1
```